



**TUGAS AKHIR - SM 141501**

**OPTIMASI PENJADWALAN PENGGUNAAN  
RUANG OPERASI MENGGUNAKAN  
ALGORITMA GENETIKA (STUDI KASUS RSUD  
DR. SOETOMO DIVISI BEDAH ORTOPEDI)**

**LAILATUL IZZA  
NRP 1212 100 023**

**Dosen Pembimbing  
Prof. DR. Mohammad Isa Irawan, MT  
Drs. Inu Laksito W., M.I.komp**

**JURUSAN MATEMATIKA  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016**





FINAL PROJECT - SM 141501

***OPTIMIZATION FOR SCHEDULING OF  
OPERATING ROOM USING GENETIC  
ALGORITHM (CASE STUDY ORTHOPEDICS  
SURGERY DIVISION OF RSUD DR. SOETOMO)***

LAILATUL IZZA  
NRP 1212 100 023

Supervisors  
Prof. DR. Mohammad Isa Irawan, MT  
Drs. Inu Laksito W., M.I.komp

DEPARTMENT OF MATHEMATICS  
Faculty of Mathematics and Natural Science  
Sepuluh Nopember Institute of Technology  
Surabaya 2016



## LEMBAR PENGESAHAN

### OPTIMASI PENJADWALAN PENGGUNAAN RUANG OPERASI MENGUNAKAN ALGORITMA GENETIKA (STUDI KASUS RSUD DR. SOETOMO DIVISI BEDAH ORTOPEDI)

### OPTIMIZATION FOR SCHEDULING OF OPERATING ROOM USING GENETIC ALGORITHM (CASE STUDY ORTHOPEDICS SURGERY DIVISION OF RSUD DR. SOETOMO)

#### TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana Sains  
Pada Bidang Ilmu Komputer  
Program Studi S-1 Jurusan Matematika  
Fakultas Matematika dan Ilmu Pengetahuan Alam  
Institut Teknologi Sepuluh Nopember Surabaya

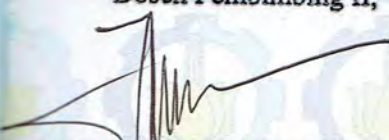
Oleh :

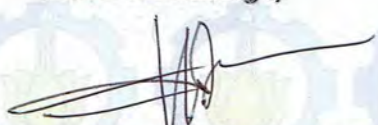
**LAILATUL IZZA**  
**NRP. 1212 100 023**

Menyetujui,

Dosen Pembimbing II,

Dosen Pembimbing I,

  
Drs. Iku Laksito W., M.I.Komp  
NIP. 19641105 198903 1 001

  
Prof. DR. Mohammad Isa Irawan, MT  
NIP. 19631225 198903 1 001



Mengetahui,  
Ketua Jurusan Matematika  
FMIPA ITS

  
Dr. Imam Mukhlash, S.Si, MT  
NIP. 19700831 199403 1 003

Surabaya, 15 Juli 2016



**OPTIMASI PENJADWALAN PENGGUNAAN  
RUANG OPERASI MENGGUNAKAN  
ALGORITMA GENETIKA (STUDI KASUS RSUD  
DR. SOETOMO DIVISI BEDAH ORTOPEDI)**

Nama Mahasiswa : Lailatul Izza  
NRP : 1212 100 023  
Jurusan : Matematika FMIPA-ITS  
Pembimbing : Prof. DR. Mohammad Isa Irawan, MT  
Drs. Inu Laksito W., M.I.Komp.

**ABSTRAK**

Sebagai rumah sakit negeri kelas A, RSUD Dr. Soetomo dijadikan sebagai rumah sakit rujukan di Jawa Timur. Dengan jumlah pasien yang tidak sedikit, RSUD Dr. Soetomo harus mampu melakukan manajemen rumah sakit. Salah satu bentuk manajemen yang dapat dilakukan adalah membuat sistem penjadwalan ruang operasi. Penjadwalan ruang operasi di RSUD Dr. Soetomo khususnya divisi bedah ortopedi masih dilakukan secara manual. Hal ini memerlukan waktu yang cukup lama. Sehingga perlu adanya sistem penjadwalan ruang operasi. Salah satu metode yang dapat digunakan untuk melakukan penjadwalan adalah algoritma genetika. Algoritma genetika membangkitkan solusi-solusi dalam bentuk kromosom yang dievaluasi menggunakan fungsi *fitness* untuk memperoleh solusi yang terbaik. Untuk melakukan penjadwalan, algoritma genetika memerlukan beberapa parameter yaitu ukuran populasi, peluang *crossover*, dan peluang mutasi. Hasil pengujian yang telah dilakukan pada sistem diperoleh bahwa sistem menghasilkan jadwal yang optimal.

**Kata-kunci** : Penjadwalan Ruang Operasi, Algoritma Genetika





**OPTIMIZATION FOR SCHEDULING OF  
OPERATING ROOM USING GENETIC ALGORITHM  
(CASE STUDY ORTHOPEDICS SURGERY DIVISION  
OF RSUD DR. SOETOMO)**

*Name of Student* : Lailatul Izza  
*NRP* : 1212 100 023  
*Department* : Mathematics of FMIPA-ITS  
*Supervisor* : Prof. DR. Mohammad Isa Irawan, MT  
Drs. Inu Laksito W., M.I.Komp.

**ABSTRACT**

*As hospital class A, Dr. Soetomo Hospital serve as a referral hospital in East Java. With patients loads, Dr. Soetomo Hospital should be able to manage the resource. One form of management that can be done is to make the scheduling system of operating room. Scheduling of the operating room at Dr. Soetomo Hospital especially the division of orthopedic surgery is still done manually. It requires quite a long time. So, Dr. Soetomo Hospital require for making the scheduling system of operating room. One method that can be used for scheduling is a genetic algorithm. Genetic algorithms generate solutions in the form of chromosomes that evaluated using the fitness function to obtain the best solution. To perform scheduling, genetic algorithm requires several parameters including population size, crossover probability, and mutation probability. The result of testing is scheduling system of operating room can make the optimal schedule.*

**Keywords** : Scheduling of the operating room, Genetic algorithms



## KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillahirobbil'aalamiin, segala puji dan syukur penulis panjatkan kehadiran Allah SWT yang telah memberikan limpahan rahmat, petunjuk serta hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul

**“OPTIMASI PENJADWALAN PENGGUNAAN  
RUANG OPERASI MENGGUNAKAN  
ALGORITMA GENETIKA (STUDI KASUS RSUD  
DR. SOETOMO DIVISI BEDAH ORTOPEDI)”**

sebagai salah satu syarat kelulusan Program Sarjana Jurusan Matematika FMIPA Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Tugas Akhir ini dapat diselesaikan dengan baik berkat bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terima kasih kepada:

1. Dr. Imam Mukhlash, S.Si, MT selaku Ketua Jurusan Matematika FMIPA ITS.
2. Prof. DR. Mohammad Isa Irawan, MT dan Drs. Inu Laksito W., M.I.Komp selaku pembimbing Tugas Akhir atas segala bimbingan, arahan, dan motivasinya kepada penulis dalam mengerjakan Tugas Akhir ini sehingga dapat diselesaikan dengan baik.
3. Drs. Daryono Budi Utomo, M.Si, Drs. Soetrisno, M. Ikomp, dan Dra. Titik Mudjiati, M.Si selaku dosen penguji atas saran yang telah diberikan demi perbaikan Tugas Akhir.

4. Dr. Didik Khusnul Arif, S.Si, M.Si selaku Ketua Program Studi Sarjana Matematika FMIPA ITS.
5. Dr. Iis Herisman, M.Si selaku sekretaris Kepala Program Studi Sarjana Matematika FMIPA ITS.
6. Drs. Suhud Wahyudi, M.Si selaku dosen wali yang telah memberikan arahan akademik selama penulis menempuh pendidikan di Jurusan Matematika FMIPA ITS.
7. Bapak dan Ibu dosen serta staf Jurusan Matematika FMIPA ITS yang tidak dapat penulis sebutkan satu-satu.
8. Kedua orang tua saya, Bapak Suriyadi dan Ibu Maslakhah atas dukungan dan semangat yang telah diberikan.
9. Adik saya Arif dan nenek Katiyu yang saya sayangi.
10. Teman teman seperjuangan 114 dan seluruh teman-teman angkatan 2012 yang saya sayangi.

Apabila dalam penulisan Tugas Akhir ini masih terdapat kekurangan, maka penulis mengharapkan kritik dan saran yang bersifat membangun demi kesempurnaan Tugas Akhir ini. Penulis berharap semoga Tugas Akhir ini dapat bermanfaat bagi banyak pihak.

Surabaya, 15 Juli 2016

Penulis

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN JUDUL .....	iii
LEMBAR PENGESAHAN .....	v
ABSTRAK .....	vii
ABSTRACT .....	ix
KATA PENGANTAR .....	xi
DAFTAR ISI .....	xiii
DAFTAR GAMBAR .....	xvii
DAFTAR SOURCE CODE .....	xix
DAFTAR TABEL .....	xxi
DAFTAR LAMPIRAN .....	xxiii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang Masalah .....	1
1.2 Rumusan Masalah .....	4
1.3 Batasan Masalah .....	4
1.4 Tujuan .....	5
1.5 Manfaat .....	5
1.6 Sistematika Penulisan .....	6
BAB II TINJAUAN PUSTAKA .....	9
2.1 Penelitian Sebelumnya .....	9
2.2 Algoritma Genetika .....	10
2.2.1 Teknik penyandian .....	12
2.2.2 Inisialialisasi populasi .....	12
2.2.3 Fungsi evaluasi .....	12
2.2.4 Seleksi .....	13
2.2.5 Operator genetika .....	15
2.2.6 Parameter kontrol .....	18
2.2.7 Konvergensi .....	19

2.2.8 Elitism .....	20
BAB III METODE PENELITIAN .....	23
3.1 Tahap Penelitian .....	23
3.2 Diagram Alir.....	26
BAB IV PERANCANGAN DAN IMPLEMENTASI....	27
4.1 Konsep Penjadwalan Ruang Operasi Divisi Bedah Ortopedi RSUD Dr. Soetomo .....	27
4.2 Analisis Sistem .....	30
4.2.1 Kebutuhan <i>User</i> .....	30
4.2.2 <i>Use Case Diagram</i> .....	31
4.2.3 <i>Activity Diagram</i> .....	31
4.3 Perancangan Sistem.....	33
4.3.1 Perancangan <i>Database</i> .....	33
4.3.2 Pengolahan Data.....	36
4.4 Gambaran Sistem Secara Umum .....	37
4.5 Perancangan Algoritma Genetika .....	37
4.5.1 Inisialisasi Parameter.....	37
4.5.2 Variabel yang Digunakan .....	38
4.5.3 Interpretasi Kromosom.....	41
4.5.4 Seleksi .....	44
4.5.5 <i>Crossover</i> .....	44
4.5.6 Mutasi.....	45
4.5.7 Fungsi <i>Fitness</i> .....	45
4.5.8 Fungsi Kendala.....	46
4.5.9 Fungsi Objektif.....	50
4.5.10 <i>Elitism</i> .....	51
4.6 Implementasi .....	51
4.6.1 Implementasi Antarmuka .....	51
4.6.2 Implementasi <i>Database</i> .....	56
4.6.3 Implementasi Pembentukan Kromosom.....	57
4.6.4 Implementasi Seleksi.....	57

4.6.5 Implementasi <i>Crossover</i> .....	58
4.6.6 Implementasi Mutasi.....	60
4.6.7 Implementasi Fungsi <i>Fitness</i> .....	61
4.6.8 Implementasi <i>Elitism</i> .....	62
BAB V PENGUJIAN DAN PEMBAHASAN HASIL ..	65
5.1 Pengujian Pengaruh Kombinasi Parameter Terhadap Kestabilan Nilai <i>Fitness</i> .....	65
5.2 Pengujian Pengaruh Peluang <i>Crossover</i> dan Mutasi pada Sistem Terhadap Waktu Komputasi .....	68
5.3 Pengujian Banyaknya Pasien yang Dijadwalkan .	70
BAB VI PENUTUP .....	71
6.1 Kesimpulan .....	71
6.2 Saran.....	72
DAFTAR PUSTAKA.....	73
LAMPIRAN .....	77
BIODATA PENULIS.....	105





## DAFTAR GAMBAR

<b>Gambar 1</b>	Diagram alir algoritma genetika.....	11
<b>Gambar 2</b>	Diagram alir Tugas Akhir .....	26
<b>Gambar 3</b>	Use case diagram sistem .....	31
<b>Gambar 4</b>	Activity diagram sistem .....	32
<b>Gambar 5</b>	CDM sistem .....	34
<b>Gambar 6</b>	PDM sistem.....	35
<b>Gambar 7</b>	Antarmuka data pasien yang akan dijadwalkan .....	52
<b>Gambar 8</b>	Antarmuka memasukkan data pasien baru...	53
<b>Gambar 9</b>	Antarmuka penjadwalan menggunakan algoritma genetika .....	54
<b>Gambar 10</b>	Contoh salah satu hasil cetak jadwal untuk ruang 610.....	56
<b>Gambar 11</b>	Rancangan tabel dan relasinya dalam DBMS MySQL.....	56



## DAFTAR SOURCE CODE

<i>Source Code 1 Method</i> untuk membentuk kromosom/individu .....	57
<i>Source Code 2 Method</i> untuk melakukan seleksi .....	58
<i>Source Code 3 Method</i> untuk melakukan <i>crossover</i> .....	58
<i>Source Code 4</i> Lanjutan <i>method</i> untuk melakukan <i>crossover</i> .....	59
<i>Source Code 5</i> Lanjutan <i>method</i> untuk melakukan <i>crossover</i> .....	60
<i>Source Code 6 Method</i> untuk melakukan mutasi .....	61
<i>Source Code 7 Method</i> untuk fungsi <i>fitness</i> .....	61
<i>Source Code 8 Method</i> untuk melakukan proses <i>elitism</i> .....	62
<i>Source Code 9</i> Lanjutan <i>method</i> untuk melakukan proses <i>elitism</i> .....	63



## DAFTAR TABEL

<b>Tabel 1</b> Variabel yang digunakan .....	38
<b>Tabel 2</b> Hasil pengujian banyaknya nilai fitness dominan yang muncul .....	66
<b>Tabel 3</b> Hasil pengujian pengaruh peluang crossover dan mutasi pada sistem terhadap waktu komputasi.....	69
<b>Tabel 4</b> Hasil pengujian banyaknya pasien yang dijadwalkan .....	70



## DAFTAR LAMPIRAN

A. Source Code.....	77
B. Hasil Penjadwalan .....	91





# **BAB I**

## **PENDAHULUAN**

Pada bab ini dijelaskan hal-hal yang melatarbelakangi munculnya permasalahan yang dibahas dalam Tugas Akhir ini. Kemudian permasalahan tersebut disusun kedalam suatu rumusan masalah. Selanjutnya dijabarkan juga batasan masalah untuk mendapatkan tujuan yang diinginkan serta manfaat yang dapat diperoleh. Adapun sistematika penulisan Tugas Akhir ini diuraikan di bagian akhir bab ini.

### **1.1 Latar Belakang Masalah**

Rumah Sakit Umum Daerah (RSUD) Dr. Soetomo merupakan rumah sakit milik pemerintah yang terletak di Surabaya, Jawa Timur. Sebagai salah satu rumah sakit negeri kelas A, RSUD Dr. Soetomo ditetapkan sebagai rujukan rumah sakit tertinggi atau disebut sebagai rumah sakit pusat untuk wilayah Jawa Timur. Rata-rata jumlah pasien setiap tahun mencapai 675.980 pasien (RSU). Jumlah pasien yang banyak ini memiliki berbagai keluhan. Diantaranya ada yang memerlukan penanganan ringan dan ada juga yang memerlukan penanganan secara khusus. Salah satu bentuk penanganan khusus yang diberikan di RSUD Dr. Soetomo adalah operasi atau pembedahan. Pembedahan yang ditangani ada bermacam-macam salah satunya adalah bedah ortopedi.

Bedah ortopedi menangani kasus yang berhubungan dengan tulang dan persendian. Kelainan pada tulang maupun persendian ini disebabkan oleh beberapa faktor seperti faktor usia, kecelakaan akibat aktivitas sehari-hari,

aktivitas olahraga, maupun kecelakaan lalu lintas. Tingginya tingkat kecelakaan yang terjadi menyebabkan jumlah pasien meningkat. Meningkatnya jumlah pasien terutama yang memerlukan penanganan khusus (operasi/bedah) menimbulkan antrian pasien bedah yang semakin panjang. Panjangnya antrian pasien juga disebabkan oleh jumlah ruang operasi yang terbatas.

Untuk menyelesaikan permasalahan tersebut, diperlukan usaha untuk meningkatkan efisiensi penggunaan ruang operasi dengan memaksimalkan sumber daya yang ada. Salah satu usaha yang dapat dilakukan adalah dengan mengoptimalkan penjadwalan penggunaan ruang operasi untuk meningkatkan kualitas pelayanan rumah sakit tanpa mengabaikan prosedur dan etika medis. Penjadwalan penggunaan ruang operasi di divisi bedah ortopedi sendiri masih dilakukan secara manual. Sehingga memerlukan waktu yang cukup lama. Oleh karena itu, perlu dibuat program sistem penjadwalan penggunaan ruang operasi untuk menghemat waktu yang ada.

Terdapat beberapa penelitian mengenai penjadwalan penggunaan ruang operasi yang dilakukan sebelumnya, diantaranya penelitian yang dilakukan oleh Zhaoxia Zhao dan Xueping Li yaitu mengenai penjadwalan operasi elektif menggunakan *Mixed Integer Nonlinear Programming* (MINLP) dan *Constraint Programming* (CP) (Zhao, et al., 2014). Selain itu, juga terdapat penelitian yang dilakukan oleh Wei Xiang, Jiao Yin, dan Gino Lim yaitu mengenai penjadwalan penggunaan ruang operasi menggunakan *Ant Colony Optimization* (ACO) (Xiang, et al., 2015).

Pada penelitian ini digunakan algoritma genetika untuk menjadwalkan penggunaan ruang operasi. Algoritma genetika merupakan suatu teknik pencarian solusi dengan menggunakan prinsip seleksi alam. Algoritma genetika dimulai dengan menentukan himpunan penyelesaian (populasi) yang direpresentasikan dengan kromosom. Solusi dari suatu populasi diambil untuk membentuk populasi baru, dimana pemilihannya tergantung dari nilai *fitness* yang merupakan fungsi evaluasi kromosom. Hal ini diharapkan agar populasi baru yang terbentuk lebih baik dari populasi terdahulu. Proses ini dilakukan berulang-ulang sampai kondisi tertentu terpenuhi.

Algoritma genetika sudah banyak digunakan dalam melakukan penjadwalan, diantaranya yaitu untuk menyelesaikan *flowshop scheduling problem* (Chang, et al., 2008) dan (Firmansyah, et al., 2016), menjadwalkan *job-shop* (Li, et al., 2010), dan menjadwalkan kereta api (Tormos, et al., 2008). Selain itu, algoritma genetika juga merupakan salah satu metode optimasi yang fleksibel karena hanya membutuhkan input fungsi objektif dan fungsi kendala (jika ada) serta mampu beradaptasi dengan baik sehingga menghasilkan solusi relatif lebih baik. Seperti penerapan algoritma genetika pada distribusi kapal perang angkatan laut (Hozairi, et al., 2014), penempatan armada TNI-AL (Hozairi, et al., 2012), penempatan pegawai (Muhtaromi, et al., 2015), penugasan kapal patroli TNI-AL (Pudji, et al., 2014) dan (Santoso, et al., 2014), dan menyelesaikan *Capacitated Vehicle Routing Problem* (Shahab, et al., 2016) yang dapat menghasilkan solusi yang relatif baik.

Oleh karena itu, pada penelitian ini penulis mengangkat permasalahan penjadwalan penggunaan ruang operasi tersebut yang diselesaikan menggunakan algoritma genetika. Dengan adanya penelitian ini diharapkan dapat menjadi salah satu alternatif dalam menjadwalkan penggunaan ruang operasi khususnya bagi RSUD Dr. Soetomo.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan diatas, permasalahan yang dibahas dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana merumuskan fungsi kendala penjadwalan penggunaan ruang operasi di RSUD Dr. Soetomo divisi bedah ortopedi?
2. Bagaimana perumusan algoritma genetika untuk mengoptimalkan penjadwalan penggunaan ruang operasi di RSUD Dr. Soetomo divisi bedah ortopedi?
3. Bagaimana merancang dan membuat perangkat lunak penjadwalan penggunaan ruang operasi di RSUD Dr. Soetomo divisi bedah ortopedi dengan menggunakan bahasa pemrograman java?

## **1.3 Batasan Masalah**

Batasan masalah pada Tugas Akhir ini adalah sebagai berikut:

1. Penjadwalan penggunaan ruang operasi yang dilakukan hanya untuk tipe operasi elektif dengan mempertimbangkan urgensi kasus, operator yang bertugas, kelas prioritas, dan *length of stay* pasien.

2. Sistem ini tidak mempertimbangkan penambahan alokasi waktu operasi.
3. Pasien yang dijadwalkan dalam kondisi siap.
4. Tidak adanya bencana alam maupun gangguan dari luar selama waktu operasi.
5. Ruang operasi yang digunakan dalam kondisi siap.
6. Peralatan yang dibutuhkan siap.
7. Ada 3 ruang yang digunakan dalam sistem ini (hal ini disesuaikan dengan kondisi di rumah sakit).

#### **1.4 Tujuan**

Tujuan dari penulisan Tugas Akhir ini adalah sebagai berikut:

1. Merumuskan fungsi kendala penjadwalan penggunaan ruang operasi di RSUD Dr. Soetomo divisi bedah ortopedi.
2. Merumuskan algoritma genetika untuk mengoptimalkan penjadwalan penggunaan ruang operasi di RSUD Dr. Soetomo divisi bedah ortopedi.
3. Merancang dan membuat perangkat lunak penjadwalan penggunaan ruang operasi di RSUD Dr. Soetomo divisi bedah ortopedi dengan menggunakan bahasa pemrograman java.

#### **1.5 Manfaat**

Manfaat dari penulisan Tugas Akhir ini adalah sebagai berikut :

1. Memberikan pengetahuan bagi pembaca mengenai aplikasi algoritma genetika khususnya untuk penjadwalan penggunaan ruang operasi.

2. Memberikan alternatif baru bagi RSUD Dr. Soetomo divisi bedah ortopedi dalam hal pengoptimalan penjadwalan penggunaan ruang operasi.
3. Memberikan informasi untuk digunakan dalam riset selanjutnya.

## **1.6 Sistematika Penulisan**

Penulisan Tugas Akhir ini disusun dalam enam bab, yaitu:

1. **BAB I PENDAHULUAN**  
Bab ini berisi tentang gambaran umum dari penulisan Tugas Akhir yang meliputi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan.
2. **BAB II TINJAUAN PUSTAKA**  
Bab ini berisi uraian mengenai penelitian sebelumnya dan algoritma genetika.
3. **BAB III METODE PENELITIAN**  
Bab ini berisi tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir.
4. **BAB IV PERANCANGAN DAN IMPLEMENTASI SISTEM**  
Bab ini berisi tentang konsep penjadwalan, perancangan dan analisis sistem serta implementasinya.
5. **BAB V PENGUJIAN DAN PEMBAHASAN HASIL**  
Bab ini berisi tentang hasil pengujian dari sistem yang telah dibuat beserta pembahasannya.

## 6. BAB VI PENUTUP

Bab ini berisi kesimpulan akhir yang diperoleh dari Tugas Akhir serta saran untuk pengembangan penelitian selanjutnya.





## **BAB II**

### **TINJAUAN PUSTAKA**

Pada bab ini diuraikan mengenai materi yang menunjang Tugas Akhir ini diantaranya adalah penelitian sebelumnya dan algoritma genetika.

#### **2.1 Penelitian Sebelumnya**

Penelitian sebelumnya yang pernah dilakukan dalam hal penjadwalan penggunaan ruang operasi diantaranya dikerjakan oleh Zhaoxia Zhao dan Xueping Li yang melakukan penjadwalan operasi elektif menggunakan *Mixed Integer Nonlinear Programming* (MINLP) dan *Constraint Programming* (CP) (Zhao, et al., 2014). Selain itu terdapat juga penelitian yang dikerjakan oleh Wei Xiang, Jiao Yin, dan Gino Lim yang melakukan penjadwalan penggunaan ruang operasi menggunakan *Ant Colony Optimization* (ACO) (Xiang, et al., 2015).

Algoritma genetika sendiri juga banyak digunakan untuk melakukan penjadwalan. Penelitian yang dikerjakan oleh Pei-Chan Chang dkk membahas mengenai penggunaan algoritma genetika untuk menyelesaikan masalah *flowshop* (Tormos, et al., 2008). Kemudian terdapat penelitian lain yang juga membahas algoritma genetika untuk menyelesaikan masalah *job-shop* oleh Ye Li dan Yan Chen (Li, et al., 2010). Algoritma genetika juga digunakan untuk melakukan penjawalan kereta api yang diteliti oleh P. Tormos dkk (Tormos, et al., 2008).

## 2.2 Algoritma Genetika

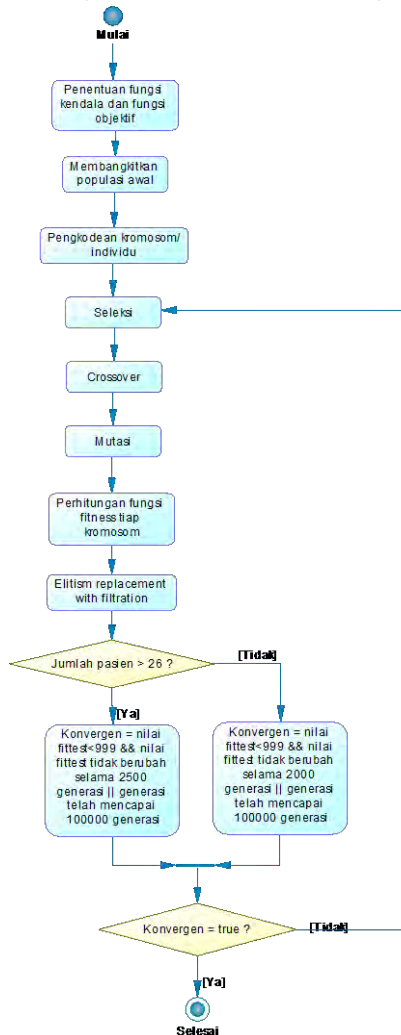
Algoritma genetika ditemukan oleh John Holland pada tahun 1960an dan dikembangkan oleh Holland pada tahun 1960an dan 1970an. Algoritma ini merupakan metode untuk menyelesaikan masalah optimasi yang langkahnya mengadopsi mekanisme evolusi biologis. Algoritma genetika bekerja dengan cara melakukan modifikasi pada populasi individu yang merupakan kandidat solusi secara berulang-ulang. Dalam setiap langkahnya, algoritma genetika memilih individu-individu yang akan dijadikan sebagai induk untuk menghasilkan keturunan. Individu-individu tersebut juga dapat mengalami mutasi. Setelah melalui beberapa generasi, populasi tersebut akan berevolusi menuju kondisi solusi optimal.

Seperi halnya algoritma optimasi yang lain, algoritma genetika dimulai dengan menentukan fungsi yang akan dioptimasi (fungsi objektif/*fitness*) dan variabel optimasinya. Algoritma ini akan berhenti jika telah memenuhi kondisi konvergensi. Dalam proses iterasinya, alur dari algoritma genetika diberikan pada Gambar 1.

Algoritma genetika menggunakan tiga aturan utama pada setiap iterasi untuk menghasilkan generasi baru dari populasi saat ini, yaitu:

1. Aturan seleksi untuk memilih individu sebagai induk yang berkontribusi pada terbentuknya generasi baru.
2. Aturan *crossover* untuk menggabungkan dua induk untuk membentuk keturunan pada generasi baru.

3. Aturan mutasi yang mengaplikasikan perubahan acak pada individu induk untuk membentuk keturunan. (Sivanandam, et al., 2008)



Gambar 1 Diagram alir algoritma genetika

Komponen-komponen utama yang ada dalam algoritma genetika adalah teknik penyandian, inisialisasi populasi, fungsi evaluasi, seleksi, operator genetika dan parameter kontrol.

### **2.2.1 Teknik penyandian**

Teknik penyandian meliputi penyandian gen dan kromosom. Gen merupakan bagian dari kromosom, satu gen biasanya mewakili satu variabel. Gen dapat direpresentasikan dalam bentuk: string bit, pohon, array bilangan real, elemen permutasi, atau representasi yang lainnya. Sehingga kromosom dapat direpresentasikan dengan:

String bit : 10011, 01101, 11101, dst.

Bilangan real : 65.65, -67.98, 562.88, dst.

Elemen permutasi : E2, E10, E5, dst.

(Kusumadewi, et al., 2005)

### **2.2.2 Inisialisasi populasi**

Ukuran populasi tergantung pada masalah yang akan dipecahkan dan jenis operator genetika yang akan diimplementasikan. Setelah menentukan ukuran populasi, kemudian dilakukan inisialisasi individu/kromosom. Inisialisasi kromosom dilakukan secara acak, namun harus tetap memperhatikan domain solusi dan kendali permasalahan yang ada. (Kusumadewi, et al., 2005)

### **2.2.3 Fungsi evaluasi**

Dalam melakukan evaluasi kromosom terdapat dua hal yang harus diperhatikan yaitu evaluasi fungsi objektif dan konversi fungsi objektif kedalam fungsi *fitness*. Secara

umum, fungsi *fitness* diturunkan dari fungsi objektif dengan nilai yang tidak negatif. Jika fungsi objektif memiliki nilai negatif, maka perlu ditambahkan suatu konstanta C sehingga nilai *fitness* yang terbentuk menjadi tidak negatif. (Kusumadewi, et al., 2005)

#### 2.2.4 Seleksi

Dari kromosom-kromosom yang ada dalam suatu populasi, perlu dipilih kromosom-kromosom yang akan dipilih untuk melakukan rekombinasi sehingga menghasilkan keturunan. Terdapat beberapa metode seleksi yang dapat digunakan diantaranya adalah:

1. *Random Selection*

Teknik yang digunakan dalam metode ini adalah memilih induk secara acak dari populasi yang nantinya akan melakukan rekombinasi untuk menghasilkan keturunan.

2. *Roulette wheel selection*

Metode ini merupakan metode yang paling sederhana, sering juga dikenal dengan *stochastic sampling with replacement*. Pada metode ini, kromosom-kromosom dipetakan dalam suatu segmen garis secara berurutan sedemikian hingga tiap-tiap segmen kromosom memiliki ukuran yang sama dengan ukuran *fitness*-nya. Sebuah bilangan acak dibangkitkan dan kromosom yang memiliki segmen dalam persekitaran bilangan acak tersebut akan terseleksi. Proses ini diulang hingga diperoleh sejumlah kromosom yang diinginkan.

3. *Stochastic universal sampling*

Pada metode ini, kromosom-kromosom dipetakan dalam suatu segmen garis secara berurutan sedemikian

hingga tiap-tiap segmen kromosom memiliki ukuran yang sama dengan ukuran *fitness*-nya seperti pada seleksi roda roulette. Kemudian diberikan sejumlah pointer sebanyak kromosom yang ingin diseleksi pada garis tersebut. Andaikan  $N$  adalah jumlah kromosom yang ingin diseleksi, maka jarak antar pointer adalah  $1/N$ , dan posisi pointer pertama diberikan secara acak pada *range*  $[0, 1/N]$ .

#### 4. *Rank selection*

Pada metode *Roulette wheel selection* akan memiliki masalah jika nilai *fitness* sangat berbeda. Jika nilai *fitness* kromosom terbaik adalah 90%, maka akan menempati 90% dari segmen garis, dan kemudian kromosom lain memiliki terlalu sedikit peluang untuk dipilih. *Rank selection* merangking kromosom dalam satu populasi. Kromosom terburuk memiliki nilai *fitness* memiliki 1 dan kromosom terbaik memiliki nilai *fitness*  $N$ . Untuk melakukan seleksi terdapat beberapa cara yaitu:

- a. Pilih sepasang kromosom secara acak. Pilih nilai acak  $R$  antara 0 dan 1. Jika  $R < r$  maka gunakan kromosom pertama sebagai induk. Jika  $R \geq r$  maka gunakan kromosom kedua sebagai induk. Ulangi langkah ini untuk memilih induk kedua. Nilai  $r$  merupakan parameter dalam metode ini.
- b. Pilih dua kromosom secara acak. Kromosom dengan evaluasi tertinggi menjadi induk. Ulangi langkah ini untuk mendapatkan induk kedua.

#### 5. *Tournament selection*

Dalam metode ini akan dipilih induk dengan selektif. Kromosom terbaik dari turnamen adalah kromosom yang memiliki nilai *fitness* tertinggi. Kromosom terbaik ini akan

ditampung terlebih dahulu, kemudian akan dilakukan seleksi lagi sampai dengan jumlah kromosom yang diinginkan. (Sivanandam, et al., 2008)

### 2.2.5 Operator genetika

Dalam algoritma genetika, terdapat 2 operator genetika yaitu operator rekombinasi dan mutasi.

#### 1. Rekombinasi

Dalam melakukan rekombinasi terdapat rekombinasi bernilai real seperti rekombinasi diskret, rekombinasi menengah, rekombinasi garis, dan rekombinasi garis yang diperluas. Selain itu juga terdapat rekombinasi bernilai biner (*crossover*) yang terdapat beberapa cara diantaranya *single-point crossover*, *multi-point crossover*, *uniform crossover*, dan *permutation crossover*.

Pada *single-point crossover*, *crossover* dilakukan pada satu titik, dimana titik ditentukan secara acak. Gen-gen ditukar antar kromosom pada titik tersebut untuk menghasilkan kromosom baru. Misalkan terdapat 2 kromosom induk dengan panjang 12 dan terpilih titik *crossover*-nya adalah 5.

Induk 1 : 0 1 1 1 0 | 0 1 0 1 1 1 0

Induk 2 : 1 1 0 1 0 | 0 0 0 1 1 0 1

Setelah dilakukan *crossover* maka akan terbentuk kromosom baru yaitu:

Anak 1 : 0 1 1 1 0 0 0 0 1 1 0 1

Anak 2 : 1 1 0 1 0 0 1 0 1 1 1 0

Pada *multi-point crossover*, *crossover* dilakukan pada  $m$  titik, dimana titik-titik tersebut ditentukan secara acak. Gen-gen ditukar antar kromosom pada titik-titik

tersebut untuk menghasilkan kromosom baru. Misalkan terdapat 2 kromosom induk dengan panjang 12 dan misalkan  $m = 3$  serta terpilih titik *crossover*-nya adalah 2, 6, 10.

Induk 1 : 0 1 | 1 1 0 0 | 1 0 1 1 | 1 0

Induk 2 : 1 1 | 0 1 0 0 | 0 0 1 1 | 0 1

Setelah dilakukan *crossover* maka akan terbentuk kromosom baru yaitu:

Anak 1 : 0 1 0 1 0 0 1 0 1 1 0 1

Anak 2 : 1 1 1 1 0 0 0 0 1 1 1 0

Pada *uniform crossover*, setiap titik memiliki potensi sebagai tempat *crossover*. Sebuah mask *crossover* dibuat sepanjang panjang kromosom secara acak. Mask ini digunakan untuk menggantikan gen tertentu dari induknya, dengan kata lain induk tersebut tidak menyumbangkan gennya ke anaknya. Induk yang akan menyumbangkan gennya ke anaknya dipilih secara acak dengan probabilitas yang sama. Anak akan diwarisi gen dari induknya jika mask *crossover* bernilai 1 atau sebaliknya. Misalkan diberikan 2 kromosom induk dengan panjang 12 dengan aturan anak 1 akan diwarisi gen dari induk 1 jika mask *crossover* bernilai 1 dan akan diwarisi gen dari induk 2 jika mask *crossover* bernilai 0. Sedangkan untuk anak 2 berlaku sebaliknya.

Induk 1 : 0 1 1 1 0 0 1 0 1 1 1 0

Induk 2 : 1 1 0 1 0 0 0 0 1 1 0 1

Dengan mask *crossover*:

Mask 1 : 1 0 0 1 1 1 0 0 1 1 0 1

Mask 2 : 0 1 1 0 0 0 1 1 0 0 1 0

Setelah dilakukan *crossover* maka akan terbentuk kromosom baru yaitu:



Anak 1 : 0 1 0 1 0 0 0 0 1 1 0 0

Anak 2 : 1 1 1 1 0 0 1 0 1 1 1 1

Pada *permutation crossover*, kromosom-kromosom anak diperoleh dengan cara memilih sub-barisan gen dari satu induk dengan memperhatikan urutan dan posisi gen. (Kusumadewi, et al., 2005)

Misalkan terdapat 2 kromosom induk yaitu:

Induk 1 : 1 2 3 4 5 6 7 8 9

Induk 2 : 4 5 3 1 8 7 6 9 2

Kemudian dipilih sub-barisan gen ketiga:

Induk 1 : 1 2 3 | 4 5 6 7 8 9

Induk 2 : 4 5 3 | 1 8 7 6 9 2

Kemudian akan diperoleh kromosom baru yaitu:

Anak 1 : 1 2 3 4 5 6 7 8 9

dan

Anak 2 : 4 5 3 1 2 8 7 6 9

Pada penelitian ini, penulis menggunakan *permutation crossover* sebagai operator *crossover*. Hal ini disesuaikan dengan penelitian yang dilakukan yaitu mengenai penjadwalan penggunaan ruang operasi dengan kromosom yang digunakan adalah elemen permutasi, berisi urutan pasien yang akan menggunakan ruang operasi.

## 2. Mutasi

Setelah mengalami proses rekombinasi, kromosom baru dapat dilakukan mutasi. Mutasi ini berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi yang memungkinkan munculnya kembali gen yang tidak muncul pada inisialisasi populasi. Dalam operator mutasi terdapat 2 jenis mutasi yaitu mutasi bernilai real dan mutasi biner.

Mutasi biner diantaranya *flipping*, *interchanging*, dan *reversing*. *Flipping* dilakukan dengan mengganti gen yang bernilai 1 menjadi 0 dan sebaliknya dengan bantuan kromosom mutasi. Jika nilai gen pada kromosom mutasi adalah 1 maka nilai gen pada anak adalah kebalikan dari gen induk dan sebaliknya. Misalkan terdapat kromosom induk dengan panjang 8 yaitu:

Induk : 1 0 1 1 0 1 0 1

Kromosom mutasi : 1 0 0 0 1 0 0 1

Anak : 0 0 1 1 1 1 0 0

*Interchanging* merupakan mutasi yang terjadi dengan cara menukar posisi antara 2 gen dalam satu kromosom, titik dilakukannya pindah silang ditentukan secara acak. Misalkan terdapat kromosom induk dengan panjang 9 yaitu: 1 0 1 1 0 1 0 1 1. Kemudian kromosom tersebut dikenakan mutasi pindah silang antara gen kelima dengan gen kesembilan sehingga diperoleh kromosom baru yaitu: 1 0 1 1 1 1 0 1 0.

*Reversing* dilakukan dengan memilih posisi acak kemudian gen setelah posisi tersebut posisinya akan dibalik untuk menghasilkan kromosom anak. Misalkan terdapat kromosom induk dengan panjang 8 yaitu: 1 0 1 1 0 1 0 0. Kemudian dipilih posisi acak yaitu 5 sehingga kromosom anak yang dihasilkan adalah 1 0 1 1 0 0 0 1. (Sivanandam, et al., 2008)

Pada penelitian ini, penulis menggunakan mutasi biner yaitu *interchanging*.

### 2.2.6 Parameter kontrol

Parameter kontrol yang digunakan dalam algoritma genetika diantaranya yaitu: ukuran populasi (popsize),

peluang *crossover* ( $p_c$ ), dan peluang mutasi ( $p_m$ ). Nilai ini ditentukan berdasarkan permasalahan yang akan dipecahkan. Beberapa rekomendasi penggunaan parameter yaitu:

- a. Untuk permasalahan yang memiliki kawasan solusi cukup besar, De Jong merekomendasikan untuk nilai parameter kontrolnya yaitu:  
 $(\text{popsize}; p_c; p_m) = (50; 0,6; 0,001)$
- b. Jika rata-rata *fitness* setiap generasi digunakan sebagai indikator, maka Grefenstette merekomendasikan untuk nilai parameter kontrolnya yaitu:  
 $(\text{popsize}; p_c; p_m) = (80; 0,95; 0,01)$
- c. Jika *fitness* dari individu terbaik dipantau pada setiap generasi, maka direkomendasikan untuk menggunakan nilai parameter kontrol:  
 $(\text{popsize}; p_c; p_m) = (80; 0,45; 0,01)$
- d. Ukuran populasi sebaiknya tidak lebih kecil dari 30 untuk sembarang jenis permasalahan.

(Kusumadewi, et al., 2005)

### 2.2.7 Konvergensi

Konvergensi merupakan kondisi pemberhentian dari proses evolusi pada algoritma genetika. Terdapat beberapa kondisi pemberhentian diantaranya adalah:

- a. Generasi maksimal  
 Proses algoritma genetika berhenti jika jumlah generasi yang telah ditentukan telah terlampaui.
- b. Waktu yang terlampaui

Proses algoritma genetika berhenti jika waktu yang telah ditentukan telah terlampaui. Dengan catatan jika jumlah generasi maksimal telah terlampaui sebelum waktu yang ditentukan terlampaui maka proses berhenti.

- c. Tidak ada perubahan pada nilai *fitness*  
Proses algoritma genetika berhenti jika tidak ada perubahan nilai *fitness* dari populasi terbaik untuk sejumlah generasi tertentu. Dengan catatan jika jumlah generasi maksimal telah terlampaui dengan tanpa adanya perubahan pada nilai *fitness* maka proses berhenti.
- d. Generasi yang gagal  
Proses algoritma genetika berhenti jika tidak ada perbaikan dalam fungsi objektif untuk beberapa generasi berturut-turut dari panjang generasi yang gagal.
- e. Batas waktu gagal  
Proses algoritma genetika berhenti jika tidak ada perbaikan dalam fungsi objektif selama selang waktu tertentu dalam detik.

(Sivanandam, et al., 2008)

### 2.2.8 Elitism

Dengan adanya proses *crossover* dan mutasi, terdapat risiko hilangnya solusi optimum karena tidak adanya jaminan bahwa operasi ini akan melindungi *fitness*. Untuk mengatasi hal itu, kromosom-kromosom terbaik dalam populasi akan diselamatkan terlebih dahulu sebelum operasi *crossover* dan mutasi dilakukan. Strategi ini dinamakan elitism. Setelah populasi yang baru terbentuk

dan dievaluasi, akan diperiksa apakah struktur populasi ini sudah mengokomodasi kromosom terbaik yang tadi diselamatkan. Jika belum, maka salinan dari kromosom tersebut akan diselipkan kembali ke populasi, biasanya untuk menggantikan individu yang paling kecil nilai *fitness*-nya. (Wati, 2010)



## **BAB III**

### **METODE PENELITIAN**

Bab ini menguraikan metode yang akan digunakan pada penelitian secara rinci. Metodologi penelitian yang digunakan berguna sebagai acuan sehingga penelitian ini dapat disusun secara sistematis.

#### **3.1 Tahap Penelitian**

Penyusunan Tugas Akhir ini dilakukan dengan menggunakan beberapa tahap yang dapat mendukung penulis dari awal hingga akhir. Dimana tahap-tahap ini akan dijadikan acuan dalam penelitian pada Tugas Akhir ini sehingga penelitian dapat disusun secara sistematis. Adapun tahap-tahap penelitian yang digunakan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

a. **Studi Literatur**

Tahap pertama dalam pengerjaan Tugas Akhir ini adalah studi literatur. Pada tahap ini akan dipelajari teori-teori yang dapat menunjang penyelesaian Tugas Akhir ini, seperti algoritma genetika yang digunakan dalam melakukan penjadwalan itu sendiri. Selain itu, pada tahap ini juga digunakan untuk mempelajari literatur-literatur ilmiah yang memiliki hubungan dengan topik penelitian yang sedang penulis lakukan.

b. **Pengumpulan Data**

Pada tahap ini, dilakukan pengumpulan data yang berkaitan dengan topik penelitian yang sedang penulis lakukan. Pengumpulan data dilakukan di RSUD Dr. Soetomo divisi bedah ortopedi. Data yang dikumpulkan oleh penulis adalah data jadwal penggunaan ruang operasi

yang meliputi data jadwal keluar masuk pasien, data dokter dan asisten dokter yang menangani pasien tersebut, data ruang operasi yang digunakan, dan data kasus yang ditangani. Selain itu, penulis juga melakukan diskusi dengan pihak RSUD Dr. Soetomo bagian penjadwalan ruang bedah ortopedi untuk mendapatkan variabel yang digunakan untuk mempertimbangkan penjadwalan penggunaan ruang bedah ortopedi.

c. Menentukan Fungsi Kendala dan Fungsi Objektif pada Penjadwalan Penggunaan Ruang Operasi

Pada tahap ini, dilakukan analisis data jadwal penggunaan ruang operasi dan data yang diperoleh dari hasil diskusi dengan pihak RSUD Dr. Soetomo bagian penjadwalan ruang bedah ortopedi. Hasil analisis ini digunakan untuk menentukan variabel yang mempengaruhi penjadwalan ruang operasi dan merumuskan fungsi kendala serta fungsi objektif pada penjadwalan ruang operasi dalam model matematika yang kemudian digunakan untuk proses evaluasi pada algoritma genetika.

d. Perancangan dan Implementasi Sistem

Pada tahap ini, dilakukan perancangan sistem penjadwalan penggunaan ruang operasi dengan mengimplementasikan algoritma genetika. Implementasi perangkat lunak dikerjakan dalam java menggunakan *compiler* Netbeans 7.3.1.

e. Pengujian dan Evaluasi Sistem

Setelah melakukan implementasi sistem kemudian dilakukan pengujian terhadap sistem yang telah dibuat. Jika masih terdapat *error* maupun adanya ketidaksesuaian



pada sistem akan dilakukan evaluasi dan perbaikan pada sistem.

f. Penarikan Kesimpulan dan Saran

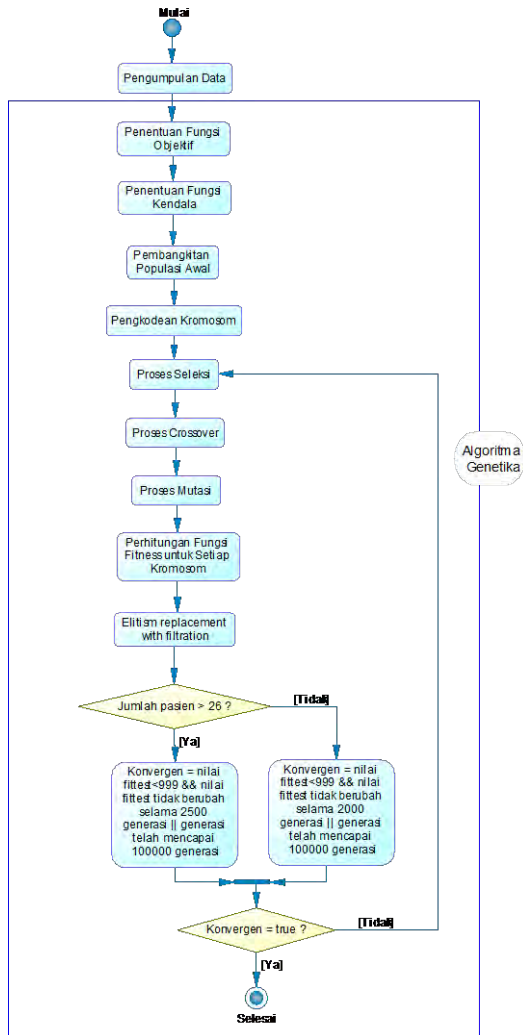
Setelah dilakukan pengujian sistem kemudian dilakukan penarikan kesimpulan dari hasil pengujian yang dilakukan. Selain itu, juga akan diberikan saran sebagai masukan dalam pengembangan penelitian selanjutnya.

g. Penulisan Tugas Akhir

Tahap terakhir yang dilakukan dalam Tugas Akhir ini adalah penulisan Tugas Akhir dan membukukan hasil penelitian yang telah dilakukan.

### 3.2 Diagram Alir

Berikut ini merupakan diagram alir dari Tugas Akhir.



Gambar 2 Diagram alir Tugas Akhir

## **BAB IV**

### **PERANCANGAN DAN IMPLEMENTASI**

Pada bab ini dijelaskan mengenai perancangan sistem dan hasil implementasi semua proses yang telah dirancang sebelumnya. Pembahasan perancangan sistem diawali dengan penjelasan tentang konsep penjadwalan penggunaan ruang operasi bedah ortopedi di RSUD Dr. Soetomo dengan menggunakan algoritma genetika, analisis sistem, dan perancangan sistem. Selanjutnya membahas mengenai implementasi sistem yang dimulai dari implementasi program pada sistem dan dilanjutkan dengan hasil implementasi antarmuka dan keseluruhan proses di dalam sistem.

#### **4.1 Konsep Penjadwalan Ruang Operasi Divisi Bedah Ortopedi RSUD Dr. Soetomo**

Sebagai rumah sakit negeri kelas A, RSUD Dr. Soetomo harus mampu memberikan pelayanan yang terbaik bagi pasiennya. Salah satu bentuk pelayanan yang dapat dilakukan adalah dengan manajemen penggunaan ruang operasi yaitu dengan cara membuat sistem penjadwalan ruang operasi. Pada Tugas Akhir ini dibahas perancangan dan pengimplementasian sistem penjadwalan ruang operasi pada divisi bedah ortopedi.

Dalam melakukan penjadwalan perlu diperhatikan ketersediaan sumber daya yang terlibat dalam penjadwalan. Deskripsi ketersediaan sumber daya yang terlibat dalam penjadwalan dari divisi bedah ortopedi sendiri yaitu sebagai berikut:

1. Jumlah spesialis yang ada di divisi bedah ortopedi (bertindak sebagai opeartor) sebanyak 15 orang yaitu TRI, MED, JUP, MOY, Prof.BP, EDY, BEE, HSO, MCH, SBS, ARN, DNU, Teri, PAY, dan KIS.
2. Jumlah PPDS divisi bedah ortopedi (bertindak sebagai operator) sekitar 55 orang.
3. Jumlah ruang operasi divisi ortopedi sebanyak 3 ruang (610, 612, 614).
4. Dalam masing-masing ruang hanya terdapat 1 tempat tidur.

Sistem penjadwalan ini dilakukan untuk mengatur urutan operasi pada pasien bedah ortopedi (hanya bedah elektif) dengan memperhatikan 5 hal yaitu:

1. Jenis kasus.  
Jenis kasus yang ditangani ini maksudnya adalah membedakan prioritas pasien yang akan melakukan operasi. Jenis kasus dibedakan menjadi 2 yaitu *urgent* dan tidak. Jika suatu kasus dikatakan *urgent*, maka sebelum 2 minggu pasien harus sudah dikerjakan.
2. Operator.  
Operator yang melakukan operasi di bedah ortopedi dibedakan menjadi 2 yaitu spesialis dan tidak. Jika pasien harus dikerjakan oleh spesialis maka perlu diperhatikan penempatan urutan operasi sehingga tidak ada pasien yang berbeda dikerjakan oleh 1 spesialis dalam waktu yang bersamaan. Jika pasien tidak harus dikerjakan oleh spesialis maka dikerjakan oleh PPDS. Untuk pasien yang dikerjakan oleh PPDS tidak perlu memperhatikan

operator yang mengerjakan karena pasien tidak harus dikerjakan oleh PPDS tertentu.

3. *Length of Stay*

*Length of stay* merupakan data lama pasien mengantri untuk melakukan operasi, terhitung dari data pasien masuk sebagai pasien yang harus dioperasi hingga hari penjadwalan dilakukan. Pasien dengan *length of stay* lebih lama didahulukan untuk dioperasi. *Length of stay* ini diperhitungkan ketika jenis kasus dari pasien sama dan operator yang mengerjakan sama.

4. Kelas prioritas.

Kelas prioritas terdiri dari kelas utama, kelas 1, kelas 2 dan kelas 3. Kelas prioritas yang lebih tinggi lebih didahulukan dibandingkan dengan kelas prioritas dibawahnya. Kelas prioritas ini diperhitungkan ketika jenis kasus dari pasien sama, operator yang mengerjakan sama, dan *length of stay*-nya sama.

5. Perkiraan lama operasi.

Perkiraan lama operasi ini digunakan untuk menentukan urutan pasien sehingga dalam 1 hari jumlah waktu operasi dan *setup time*-nya tidak melebihi waktu buka ruang operasi.

Selain itu, penjadwalan yang dilakukan juga harus memperhatikan jumlah ruang operasi yang digunakan dan siap digunakan mulai pukul 09.00 hingga pukul 16.00.

Pada Tugas Akhir ini sistem penjadwalan dibuat menggunakan algoritma genetika, dengan memperhatikan beberapa asumsi berikut:

1. Pasien yang dijadwalkan dalam kondisi siap.

2. Alat dan *implant* (jika diperlukan) sudah tersedia.
3. Operator yang dijadwalkan dalam kondisi siap.
4. Ruang operasi siap digunakan.
5. Tidak adanya faktor luar yang menghambat proses operasi yang sedang berjalan.
6. Tidak adanya penambahan waktu dalam proses operasi.
7. *Time setup* pergantian pasien selama 30 menit.

Sistem ini dibuat dengan menghasilkan jadwal yang memenuhi 5 hal di atas. Selain itu, sistem dibuat untuk meminimalkan jumlah hari operasi dan jumlah ruang operasi yang digunakan di hari terakhir penjadwalan untuk  $n$  pasien.

## 4.2 Analisis Sistem

Sistem penjadwalan ini dibuat dengan memperhatikan beberapa analisis sistem, yaitu kebutuhan *user*, *use case diagram* dari sistem, dan *activity diagram* dari sistem. 3 hal ini dijelaskan dalam subbab selanjutnya.

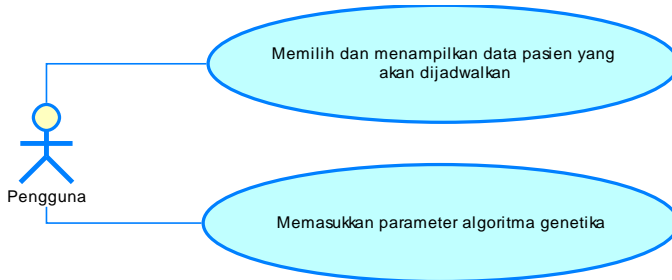
### 4.2.1 Kebutuhan User

Sistem ini dirancang untuk memenuhi kebutuhan penjadwalan penggunaan ruang operasi divisi bedah ortopedi RSUD Dr. Soetomo. Kebutuhan *user* tersebut diantaranya:

1. Dapat menjadwalkan  $n$  pasien secara otomatis.
2. Jadwal yang dihasilkan dapat dicetak maupun disimpan dalam bentuk .pdf.

#### 4.2.2 Use Case Diagram

*Use case diagram* merupakan diagram yang menggambarkan interaksi antara pengguna sistem dan sistem yang dibuat.

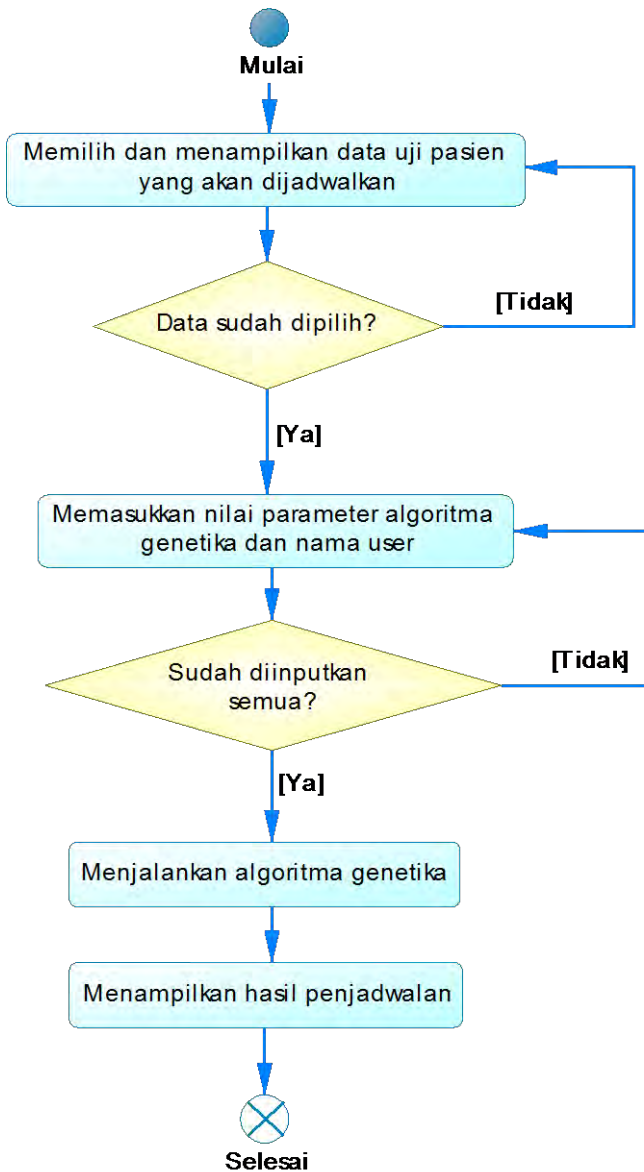


**Gambar 3** *Use case diagram* sistem

Pada gambar *use case diagram* diatas dapat dilihat bahwa pengguna dapat memilih dan menampilkan data pasien yang akan dijadwalkan, memasukkan parameter algoritma genetika. Kemudian sistem melakukan penjadwalan, menampilkan hasil penjadwalan, nilai *fitness* dari kromosom terbaik, lama *running time*, kondisi pemenuhan kendala dari kromosom terbaik, dan pencetakan hasil penjadwalan.

#### 4.2.3 Activity Diagram

*Activity diagram* merupakan diagram untuk menjelaskan alur aktifitas yang terdapat dalam sistem yang dirancang. Diagram ini dirancang untuk data uji dengan parameter dimasukkan oleh *user*.



Gambar 4 Activity diagram sistem



Pada gambar *activity diagram* diatas, sistem dimulai dengan memilih dan menampilkan data uji pasien yang akan dijadwalkan. Jika belum ada data yang dipilih maka akan kembali untuk memilih data uji. Apabila sudah dipilih maka selanjutnya yaitu memasukkan nilai parameter algoritma genetika diantaranya ukuran populasi, peluang *crossover*, dan peluang mutasi serta nama *user*. Jika salah satu parameter atau nama *user* belum diisi maka akan kembali pada aktivitas untuk memasukkan nilai parameter algoritma genetika dan nama *user*. Apabila semua nilai parameter algoritma genetika dan nama *user* telah diisi, maka sistem akan menjalankan algoritma genetika. Kemudian sistem akan menampilkan hasil penjadwalannya.

### **4.3 Perancangan Sistem**

Sistem ini dirancang dan dibangun menggunakan *software* Netbeans 7.3.1 dengan *database management system* (DBMS) yang digunakan adalah MySQL.

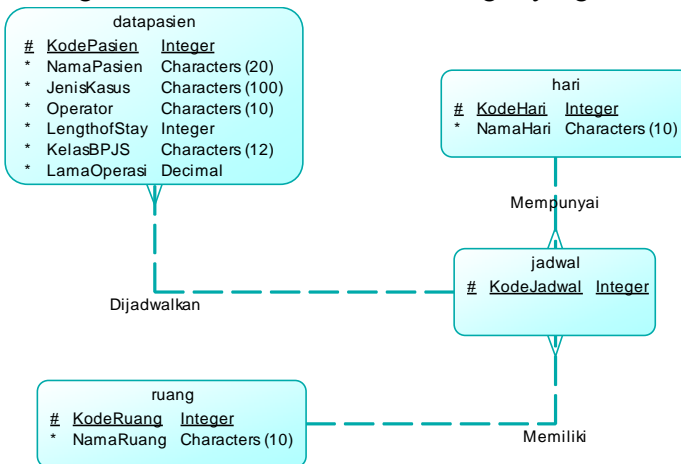
#### **4.3.1 Perancangan Database**

Sistem penjadwalan penggunaan ruang operasi ini dibuat dengan menggunakan *database management system* (DBMS) yang digunakan adalah MySQL. Rancangan *database*-nya sendiri dijelaskan dalam subbab berikutnya yang meliputi CDM dan PDM.

##### **4.3.1.1 Conceptual Data Model (CDM)**

*Conceptual Data Model* (CDM) merupakan tahapan untuk menganalisis struktur konsep dari suatu sistem, yang meliputi identifikasi entitas-entitas yang diperlukan dalam

sistem, atribut-atribut untuk masing-masing entitas, dan hubungan atau relasi antara entitas 1 dengan yang lain.

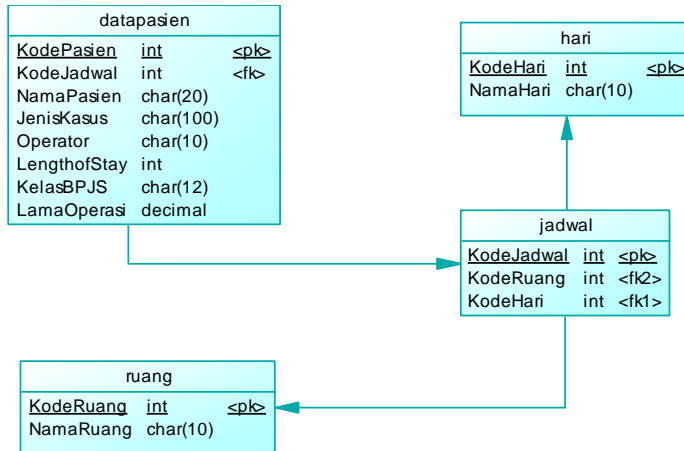


**Gambar 5** CDM sistem

Pada gambar diatas terlihat bahwa terdapat 3 entitas yang diperlukan untuk menyimpan data dalam sistem penjadwalan ini. Entitas yang pertama yaitu entitas datapasien, hari, jadwal, dan ruang. Untuk entitas datapasien memiliki atribut yaitu KodePasien, NamaPasien, JenisKasus, Operator, *LengthofStay*, KelasBPJS, dan LamaOperasi. Sedangkan untuk entitas hari memiliki atribut KodeHari dan NamaHari, entitas jadwal memiliki atribut KodeJadwal. Kemudian untuk entitas ruang memiliki atribut KodeRuang dan NamaRuang. Relasi yang terbentuk dari entitas datapasien ke jadwal yaitu *many to one*. Relasi yang terbentuk dari entitas jadwal ke hari dan relasi antara entitas jadwal ke ruang yaitu *many to one*. Dari hasil CDM ini kemudian di-*generate* ke dalam bentuk PDM.

#### 4.3.1.2 Physical Data Model (PDM)

*Physical Data Model* (PDM) merupakan tahapan untuk menganalisis tabel yang digunakan untuk sistem. Tabel ini diperoleh dari pengolahan entitas dan relasi dari *Conceptual Data Model* (CDM).



**Gambar 6** PDM sistem

Pada gambar diatas terlihat bahwa tabel yang diperlukan untuk membuat sistem penjadwalan sebanyak 4 tabel, yaitu tabel *datapasien*, *hari*, *jadwal*, dan *ruang*. Untuk *field* dari masing-masing tabel sama dengan atribut pada entitas, namun untuk tabel *datapasien* terdapat penambahan *field* yaitu *KodeJadwal* yang merupakan *foreign key* dari hasil relasi dengan entitas *jadwal*. Kemudian untuk tabel *jadwal* terdapat penambahan *field* yaitu *KodeRuang* dan *KodeHari* yang merupakan *foreign key* dari hasil relasi dengan entitas *hari* dan *ruang*.

### 4.3.2 Pengolahan Data

Data yang digunakan sistem penjadwalan ini dibagi menjadi 3 macam, yaitu data masukan, data proses, dan data keluaran. Data masukan merupakan data yang akan diproses oleh sistem. Data proses merupakan representasi data masukan ke dalam sistem. Sedangkan data keluaran merupakan data yang dihasilkan oleh sistem.

#### 4.3.2.1 Data Masukan

Data masukan yang diberikan ke sistem adalah data pasien operasi. Data ini diperoleh dari RSUD Dr. Soetomo divisi bedah ortopedi. Data yang dimasukkan terdiri atas nama pasien, jenis kasus, operator, *length of stay*, kelas prioritas, dan perkiraan lama operasi.

#### 4.3.2.2 Data Proses

Data proses diperoleh dari transformasi dari data masukan ke dalam algoritma genetika. Data masukan akan dibuat untuk membentuk kromosom dan menghitung nilai *fitness*. Data masukan yang ada akan direpresentasikan dalam bentuk angka. Data jenis kasus direpresentasikan menjadi 0 atau 1. Data operator direpresentasikan menjadi 0 sampai dengan 13. Data *length of stay* tetap. Data kelas prioritas direpresentasikan menjadi 0 sampai dengan 3. Data perkiraan lama operasi tetap.

#### 4.3.2.3 Data Keluaran

Data keluaran yang diberikan oleh sistem adalah jadwal penggunaan ruang operasi divisi bedah ortopedi dengan ruang operasi yang ada berjumlah 3 ruang (610,612,614). Jadwal yang dihasilkan terdiri atas nama pasien, hari operasi, operator, dan perkiraan lama operasi

yang ditampilkan dalam tabel di antarmuka. Jadwal ini juga dapat dicetak dengan ukuran kertas A4 maupun disimpan dalam *file* bertipe .pdf, .doc, atau .xls dengan informasi data yang lebih lengkap.

#### 4.4 Gambaran Sistem Secara Umum

Tahapan yang dilakukan oleh sistem penjadwalan ini antara lain:

1. Tahap inisialisasi parameter algoritma genetika yang meliputi ukuran populasi, peluang *crossover*, dan peluang mutasi.
2. Tahap membangkitkan kromosom yang merepresentasikan solusi.
3. Tahap *crossover* untuk menghasilkan kromosom baru sebanyak ukuran populasi.
4. Tahap mutasi pada kromosom baru yang terbentuk.
5. Tahap evaluasi untuk masing-masing kromosom dengan menggunakan fungsi *fitness*.
6. Tahap pembentukan populasi baru dengan menyaring kromosom terbaik dari hasil evaluasi sejumlah ukuran populasi (*elitism*).

#### 4.5 Perancangan Algoritma Genetika

Pada subbab ini akan dijelaskan bagaimana merancang algoritma genetika untuk studi kasus penjadwalan penggunaan ruang operasi ini.

##### 4.5.1 Inisialisasi Parameter

Inisialisasi parameter merupakan pendefinisian nilai parameter algoritma genetika. Parameter yang diinisialisasi yaitu ukuran populasi, peluang *crossover*, dan peluang mutasi. Ukuran populasi digunakan untuk menentukan

jumlah kromosom dalam 1 populasi. Dengan kata lain, ukuran populasi ini menentukan jumlah solusi untuk menyelesaikan masalah penjadwalan ini untuk setiap iterasi dalam algoritma genetika.

Peluang *crossover* digunakan untuk menentukan peluang apakah kromosom dikenai operator *crossover* atau tidak. Peluang *crossover* ini mempunyai nilai antara 0-1. Peluang mutasi digunakan untuk menentukan peluang kromosom dikenai operator mutasi. Peluang mutasi ini juga mempunyai nilai antara 0-1.

#### 4.5.2 Variabel yang Digunakan

Variabel yang digunakan untuk mendefinisikan algoritma genetika penjadwalan ruang operasi adalah sebagai berikut:

**Tabel 1** Variabel yang digunakan

<b>Variabel</b>	<b>Keterangan</b>
$jeniskasus(i)$	Nilai jenis kasus untuk pasien ke- $i$ (0,1)
$operator(i)$	Nilai operator untuk pasien ke- $i$ (0,1, ...,14)
$kelasBPJS(i)$	Nilai kelas prioritas untuk pasien ke- $i$ (0,1,2,3)
$lengthofstay(i)$	Nilai length of stay untuk pasien ke- $i$
$lamaoperasi(i)$	Waktu yang diperlukan untuk mengoperasi pasien ke- $i$
$a_i$	Pasien urutan ke- $i$ dari solusi jadwal pasien

$jeniskasus(a_i)$	Nilai jenis kasus untuk pasien urutan ke- $i$ dari solusi jadwal pasien (0,1)
$operator(a_i)$	Nilai operator untuk pasien urutan ke- $i$ dari solusi jadwal pasien (0,1, ...,14)
$bpjs(a_i)$	Nilai kelas prioritas untuk pasien urutan ke- $i$ dari solusi jadwal pasien (0,1,2,3)
$los(a_i)$	Nilai length of stay untuk pasien urutan ke- $i$ dari solusi jadwal pasien
$lamaop(a_i)$	Waktu yang diperlukan untuk mengoperasi pasien urutan ke- $i$ dari solusi jadwal pasien
$ruang(a_i)$	Ruang yang digunakan oleh pasien urutan ke- $i$ dari solusi jadwal pasien
$sesiOp(a_i)$	Sesi operasi pasien urutan ke- $i$ dari solusi jadwal pasien
$f(k_l)$	Nilai fitness kromosom ke- $l$
$u(a_i)$	Kesalahan penjadwalan yang melanggar aturan urgensitas untuk pasien urutan ke- $i$ dari solusi jadwal pasien
$o(a_i)$	Kesalahan penjadwalan yang melanggar aturan operator untuk pasien urutan ke- $i$ dari solusi jadwal pasien

$klos(a_i)$	Kesalahan penjadwalan yang melanggar aturan length of stay untuk pasien urutan ke- $i$ dari solusi jadwal pasien
$kbpjs(a_i)$	Kesalahan penjadwalan yang melanggar aturan kelas prioritas untuk pasien urutan ke- $i$ dari solusi jadwal pasien
$hariOp(a_i)$	Pasien ke- $i$ dari solusi jadwal pasien dioperasi pada hari keberapa terhitung dari hari pembuatan jadwal
$F(k_i)$	Nilai fungsi objektif dari kromosom ke- $i$
$hariOp(i, a_{n-1})$	Jumlah hari yang digunakan untuk mengoperasi $n$ pasien pada solusi kromosom ke- $i$
$ruang(i, a_{n-1})$	Jumlah ruang yang digunakan di hari terakhir operasi pada solusi kromosom ke- $i$
$ruang(a_{n-1})$	Jumlah ruang yang digunakan di hari terakhir operasi
$F(k_i)$	Nilai fungsi objektif dari kromosom ke- $i$
$hariOp(i, a_{n-1})$	Jumlah hari yang digunakan untuk mengoperasi $n$ pasien pada solusi kromosom ke- $i$
$ruang(i, a_{n-1})$	Jumlah ruang yang digunakan di hari terakhir operasi pada solusi kromosom ke- $i$



### 4.5.3 Interpretasi Kromosom

Dalam algoritma genetika kromosom merupakan representasi solusi. Dalam masalah penjadwalan penggunaan ruang operasi ini kromosom didefinisikan sebagai kromosom permutasi. Karena kromosom ini merepresentasikan solusi urutan pasien yang dioperasi.

Misalkan dalam masalah penjadwalan pada Tugas Akhir ini terdapat 6 pasien yang akan dioperasi yaitu:

1. Tn. M. Said dengan jenis kasus amputasi setinggi bahu (S) dengan operator MED, kelas prioritas kelas 2, *length of stay*-nya 25 hari, dan perkiraan lama operasinya yaitu 6 jam.
2. Tn. Darmawan dengan jenis kasus PSR dengan operator PPDS, kelas prioritas kelas 2, *length of stay*-nya 18 hari, dan perkiraan lama operasinya yaitu 1 jam.
3. An. Ahmad Arif dengan jenis kasus Disarticulasi hip (D) dengan operator MED, kelas prioritas kelas 3, *length of stay*-nya 9 hari, dan perkiraan lama operasinya yaitu 2 jam.
4. Ny. Siti Kodijah dengan jenis kasus aff plate fibula dengan operator PPDS, kelas prioritas kelas 3, *length of stay*-nya 17 hari, dan perkiraan lama operasinya yaitu 3 jam.
5. Tn. Ahmad Yajid dengan jenis kasus debridement + re ORIF dengan operator EDY, kelas prioritas kelas utama, *length of stay*-nya 25 hari, dan perkiraan lama operasinya yaitu 2 jam.
6. Lukman H. dengan jenis kasus FFMT dengan operator HSO, kelas prioritas kelas 2, *length of stay*-

nya 20 hari, dan perkiraan lama operasinya yaitu 6 jam.

Dari 6 data ini kemudian diolah menjadi bentuk kromosom, dengan informasi gen sebagai berikut:

1. Gen 0 merupakan representasi dari Tn. M. Said. Gen 0 ini mengandung informasi:  
 $jeniskasus(0) = 0, operator(0) = 1,$   
 $kelasBPJS(0) = 2, lengthofstay(0) = 25,$   
 dan  $lamaoperasi(0) = 6.$
2. Gen 1 merupakan representasi dari Tn. Darmawan. Gen 1 ini mengandung informasi:  
 $jeniskasus(1) = 0, operator(1) = -1,$   
 $kelasBPJS(1) = 2, lengthofstay(1) = 18,$   
 dan  $lamaoperasi(1) = 1.$
3. Gen 2 merupakan representasi dari An. Ahmad Arif. Gen 2 ini mengandung informasi:  
 $jeniskasus(2) = 1, operator(2) = 1,$   
 $kelasBPJS(2) = 3, lengthofstay(2) = 9,$   
 dan  $lamaoperasi(2) = 2.$
4. Gen 3 merupakan representasi dari Ny. Siti Kodijah. Gen 3 ini mengandung informasi:  
 $jeniskasus(3) = 0, operator(3) = -1,$   
 $kelasBPJS(3) = 3, lengthofstay(3) = 17,$   
 dan  $lamaoperasi(3) = 3.$
5. Gen 4 merupakan representasi dari Tn. Ahmad Yajid. Gen 4 ini mengandung informasi:  
 $jeniskasus(4) = 0, operator(4) = 5,$   
 $kelasBPJS(4) = 0, lengthofstay(4) = 25,$   
 dan  $lamaoperasi(4) = 2.$

6. Gen 5 merupakan representasi dari Lukman H. Gen 5 ini mengandung informasi:

$$\begin{aligned} \text{jeniskasus}(5) &= 0, \text{operator}(5) = 7, \\ \text{kelasBPJS}(5) &= 2, \text{lengthofstay}(5) = 20, \\ \text{dan lamaoperasi}(5) &= 6. \end{aligned}$$

Keterangan:

Informasi  $\text{jeniskasus}(i) = 0$  jika tidak *urgent*.

Informasi  $\text{jeniskasus}(i) = 1$  jika *urgent*.

Informasi  $\text{operator}(i)$  = urutan dokter, urutan dokter adalah sebagai berikut: TRI, MED, JUP, MOY, Prof.BP, EDY, BEE, HSO, MCH, SBS, ARN, DNU, Teri, PAY, dan KIS.

Informasi  $\text{operator}(i) = -1$  jika operator adalah PPDS.

Informasi  $\text{kelasBPJS}(i) = 0$  jika kelas prioritas adalah kelas utama.

Informasi  $\text{kelasBPJS}(i) = 1$  jika kelas prioritas adalah kelas 1.

Informasi  $\text{kelasBPJS}(i) = 2$  jika kelas prioritas adalah kelas 2.

Informasi  $\text{kelasBPJS}(i) = 3$  jika kelas prioritas adalah kelas 3.

Informasi  $\text{lengthofstay}(i) = \text{length of stay}$  pasien ke- $i$ .

Informasi  $\text{lamaoperasi}(i) = \text{lama operasi}$  pasien ke- $i$ .

Sehingga bentuk kromosom yang dapat dibentuk adalah sebagai berikut:

0	2	3	1	4	5
2	1	3	4	5	0
5	3	4	0	1	2

dan seterusnya, dengan masing-masing gen memiliki informasi seperti pada penjelasan diatas.

#### 4.5.4 Seleksi

Dalam algoritma genetika, seleksi digunakan untuk memilih kromosom yang akan di-*crossover*. Pada Tugas Akhir ini telah diuji 3 metode seleksi yang ada yaitu *random selection*, *roulette wheel selection*, dan *tournament selection*. Dari hasil uji diperoleh bahwa metode yang cocok untuk kasus ini adalah metode *tournament selection*. Dalam metode ini kromosom yang dipilih adalah kromosom terbaik dari kumpulan kromosom turnamen. Kromosom turnamen merupakan sekumpulan kromosom yang dipilih secara acak dari suatu populasi dengan jumlah tertentu.

#### 4.5.5 Crossover

Crossover merupakan salah satu langkah dalam algoritma genetika. *Crossover* digunakan untuk merekombinasi 2 kromosom induk dari suatu populasi sehingga menghasilkan kromosom baru sebagai generasi selanjutnya. Pada kasus ini, dalam memilih 2 kromosom induk yang melakukan *crossover* digunakan metode seleksi *tournament selection*.

Operator *crossover* yang digunakan dalam kasus ini adalah *permutation crossover* dengan mengambil 1 titik acak yang akan digunakan sebagai titik rekombinasi. Misalkan terdapat 2 kromosom A dan B.

1	3	6	5	2	4	0
6	4	3	2	1	0	5

Kemudian dibangkitkan bilangan acak untuk melihat peluang apakah kromosom A dan B dapat melakukan *crossover* atau tidak. Hal ini dilihat berdasarkan besarnya peluang *crossover* yang telah diinisialisasi. Semakin besar peluang *crossover* semakin mungkin 2 kromosom dikenakan operator *crossover*. Jika 2 kromosom tersebut dapat melakukan *crossover* maka 2 kromosom tersebut dikenakan operator *permutation crossover* dengan titik misalkan sama dengan 3. Maka akan menghasilkan 2 kromosom anak yaitu:

1	3	6	4	5	2	0
6	4	3	1	0	5	2

#### 4.5.6 Mutasi

Setelah dilakukan *crossover* dan terbentuk kromosom baru, kemudian kromosom baru ini akan dikenai operator mutasi yaitu *interchanging*. Namun tidak semua kromosom baru dikenai operator ini, tergantung besarnya peluang mutasi yang didefinisikan. Operator *interchanging* dilakukan dengan cara memilih 2 titik secara acak yang merupakan indeks dari gen dalam 1 kromosom. Kemudian akan dilakukan penukaran posisi dari kedua gen tersebut.

#### 4.5.7 Fungsi *Fitness*

Fungsi *fitness* dalam algoritma genetika merupakan representasi kebugaran suatu kromosom. Fungsi *fitness* ini dapat diperoleh dari fungsi kendala dan fungsi objektif dalam penjadwalan ruang operasi. Fungsi kendala dan fungsi objektif dari penjadwalan ruang operasi ini akan

dijelaskan pada subbab berikutnya. Berdasarkan fungsi kendala dan fungsi objektif tersebut maka fungsi *fitness* dalam algoritma genetika ini didefinisikan sebagai berikut:

$$\begin{aligned}
 f(k_l) = & \left( 1000000 \times \sum_{i=0}^{n-1} u(a_i) \right) + \left( 100000 \times \sum_{i=0}^{n-1} o(a_i) \right) \\
 & + \left( 10000 \times \sum_{i=0}^{n-1} klos(a_i) \right) \\
 & + \left( 1000 \times \sum_{i=0}^{n-1} kbpjs(a_i) \right) \\
 & + (10 \times hariOp(a_{n-1}) + ruang(a_{n-1}))
 \end{aligned}$$

dengan:

$k_l$  merupakan kromosom ke- $l$ ,  $l = 1, 2, \dots, m$ ,  $m$  merupakan ukuran populasi,  $n$  merupakan jumlah pasien.

Konstanta 1000000, 100000, 10000, 1000, 10 dipilih karena nilai *fitness* ini merepresentasikan kesalahan solusi penjadwalan yang diberikan. Dengan urutan prioritas kendala yang harus dipenuhi terlebih dahulu adalah urgensi, operator, *length of stay*, kemudian kelas prioritas. Sehingga dari satu kendala dengan kendala yang lain dibuat dengan kelipatan 10.

Kromosom yang baik adalah kromosom yang memiliki nilai *fitness* terkecil. Hal ini dikarenakan nilai *fitness* ini merepresentasikan besarnya nilai kesalahan suatu kromosom terhadap fungsi kendala dan lamanya hari penjadwalan untuk  $n$  pasien.

#### 4.5.8 Fungsi Kendala

Fungsi kendala berfungsi untuk mengontrol solusi sehingga solusi yang diberikan dapat sesuai dengan aturan

yang ada. Aturan dalam membuat jadwal penggunaan ruang operasi meliputi 4 hal yaitu jenis kasus yang dikerjakan, operator yang melakukan operasi, kelas prioritas dari pasien dan perkiraan lama operasi (penjelasan ada di subbab 4.1). Fungsi kendala untuk masing-masing aturan adalah sebagai berikut:

1. Jenis kasus.

Aturan ini digunakan untuk mengontrol supaya dalam penjadwalan, kasus *urgent* dijadwalkan sebelum 2 minggu terhitung sejak kasus masuk di rumah sakit.

Misalkan  $a_i, i = 0, 1, \dots, n - 1$  adalah urutan pasien yang akan dioperasi dan nilai urgensi untuk pasien ke- $i$  adalah  $u(a_i)$ . Dengan  $u(a_i) = 1$  jika kasus *urgent* telah melewati batas 2 minggu dan  $u(a_i) = 0$  jika kasus *urgent*/tidak masih kurang dari 2 minggu. Sehingga nilai urgensi dapat dinyatakan dengan:

$$u(a_i) = \begin{cases} 1 & ; \text{lengthofstay}(a_i) + \text{hariOp}(a_i) \geq 14 \\ 0 & ; \text{lain} \end{cases}$$

dengan:

$i = 0, 1, \dots, n - 1$ ,  $n$  adalah jumlah pasien yang dijadwalkan.

Dengan demikian solusi urutan pasien harus memenuhi:

$$\sum_{i=0}^{n-1} u(a_i) = 0$$

2. Operator.

Aturan ini digunakan untuk mengontrol supaya dalam penjadwalan tidak terjadi jadwal yang

bentrok. Dengan kata lain, tidak ada 1 spesialis yang mengerjakan 2 pasien di waktu bersamaan.

Misalkan  $a_i, i = 0, 1, \dots, n - 1$  adalah urutan pasien yang akan dioperasi dan nilai jadwal seorang pasien ke- $i$  terhadap operator (bentrok) adalah  $o(a_i)$ .  $o(a_i) = 1$  jika jadwal pasien ke- $i$  bentrok dengan pasien yang lain dan  $o(a_i) = 0$  jika jadwal pasien ke- $i$  tidak bentrok dengan pasien yang lain. Sehingga untuk aturan operator ini dapat dinyatakan dengan:

$$o(a_i) = \begin{cases} 1; & \text{jadwal pasien ke } i \text{ bentrok} \\ 0; & \text{sebaliknya} \end{cases}$$

dengan:

$i = 0, 1, \dots, n - 1$ ,  $n$  adalah jumlah pasien yang dijadwalkan.

Jadwal pasien ke- $i$  bentrok jika

$operator(a_i) = operator(a_j)$  dan

$hariOp(a_i) = hariOp(a_j)$  dan

$ruang(a_i) = ruang(a_j)$  dan

$sesiOp(a_i) = sesiOp(a_j)$ ,

dengan  $j \neq i, j = 0, 1, \dots, n - 1$ .

Dengan demikian solusi urutan pasien harus memenuhi:

$$\sum_{i=0}^{n-1} o(a_i) = 0$$

### 3. *Length of stay.*

Aturan ini digunakan untuk mengontrol penjadwalan supaya pasien yang *length of stay* lebih lama dapat didahulukan dibandingkan pasien yang



lain dengan syarat jenis kasusnya yang sama dan dikerjakan oleh operator yang sama pula.

Misalkan  $a_i, i = 0, 1, \dots, n - 1$  adalah urutan pasien yang akan dioperasi, dengan  $n$  adalah jumlah pasien.

Jika  $jeniskasus(a_i) = jeniskasus(a_j)$  dan  $operator(a_i) = operator(a_j)$ , maka nilai  $klos(a_i)$  adalah:

$$klos(a_i) = \begin{cases} 1, los(a_i) < los(a_j); j > i \\ 0, yang lain \end{cases}$$

dengan  $j = 1, \dots, n - 1$ .

Dengan demikian solusi urutan pasien harus memenuhi:

$$\sum_{i=0}^{n-1} klos(a_i) = 0$$

#### 4. Kelas prioritas.

Aturan ini digunakan untuk mengontrol penjadwalan supaya pasien yang kelas prioritas lebih tinggi dapat didahulukan dibandingkan pasien yang lain dengan syarat jenis kasusnya yang sama dan dikerjakan oleh operator yang sama serta memiliki *length of stay* yang sama.

Misalkan  $a_i, i = 0, 1, \dots, n - 1$  adalah urutan pasien yang akan dioperasi, dengan  $n$  adalah jumlah pasien.

Jika  $jeniskasus(a_i) = jeniskasus(a_j)$  dan

$operator(a_i) = operator(a_j)$  dan

$los(a_i) = los(a_j)$ , maka nilai  $kbpjs(a_i)$  adalah:

$$kbpjs(a_i) = \begin{cases} 1, bpjs(a_i) > bpjs(a_j); j > i \\ 0, yang lain \end{cases}$$

dengan  $j = 1, \dots, n - 1$ .

Dengan demikian solusi urutan pasien harus memenuhi:

$$\sum_{i=0}^{n-1} kbpjs(a_i) = 0$$

5. Perkiraan lama operasi.

Aturan ini digunakan untuk mengontrol penjadwalan supaya urutan pasien tiap harinya tidak melebihi waktu buka ruang operasi. Hal ini juga digunakan sebagai fungsi fitness dalam algoritma genetika.

Dengan waktu *ready* ruang operasi yaitu 09.00-16.00 maka terdapat 7 jam per ruang untuk melakukan operasi. Misalkan  $a_i, i = 0, 1, \dots, n - 1$  adalah urutan pasien yang akan dioperasi, dengan  $n$  adalah jumlah pasien. Maka hasil penjadwalan per ruang harus memenuhi:

$$\sum_{i=0}^{m-1} (lamaop(a_i) + 0.5) \leq 7$$

dengan 0.5 adalah waktu jeda perpindahan antara operasi 1 dengan operasi yang lain, dan  $m$  adalah jumlah pasien yang memenuhi pertidaksamaan tersebut.

#### 4.5.9 Fungsi Objektif

Fungsi objektif digunakan untuk memberikan solusi yang optimum. Fungsi objektif penjadwalan ini adalah meminimalkan hari operasi dan jumlah ruang yang digunakan di hari terakhir dalam penjadwalan  $n$  pasien.

$$F(k_i) = \text{hariOp}(i, a_{n-1}) + \text{ruang}(i, a_{n-1})$$

dengan  $i = 1, \dots, m$ ,  $m$  ukuran populasi dan  $n$  adalah jumlah pasien.

#### 4.5.10 *Elitism*

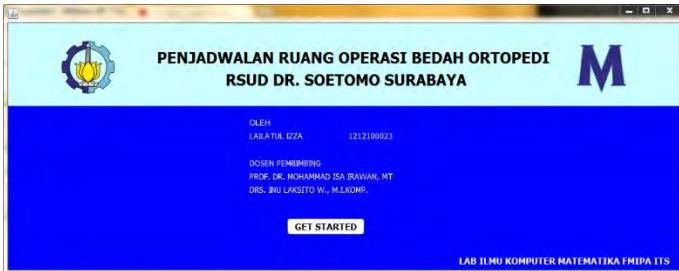
Tahap *elitism* merupakan tahap untuk memilih kromosom yang akan dipertahankan pada generasi selanjutnya. Pada Tugas Akhir ini digunakan *elitism replacement with filtration*. Metode ini berguna untuk menyaring kromosom-kromosom awal dan kromosom-kromosom hasil *crossover* dan mutasi yang memiliki nilai *fitness* terkecil. Selain itu, dengan menggunakan metode ini kromosom yang memiliki nilai *fitness* yang sama akan dihapus. Jika jumlah kromosom yang dihasilkan pada proses *elitism* kurang dari ukuran populasi maka ditambahkan kromosom baru yang dibangkitkan secara acak.

### 4.6 Implementasi

Perancangan sistem yang telah dirancang, kemudian diimplementasikan pada bahasa pemrograman java dengan menggunakan *software* Netbeans 7.3.1 dan *database management system* (DBMS) MySQL.

#### 4.6.1 Implementasi Antarmuka

Antarmuka digunakan untuk mempermudah *user* dalam penggunaan program. Antarmuka sistem penjadwalan ini terdiri dari 4 yaitu antarmuka home, antarmuka data pasien yang akan dijadwalkan, antarmuka untuk memasukkan pasien baru, dan antarmuka untuk penjadwalan menggunakan algoritma genetika.



**Gambar 7** Antarmuka home

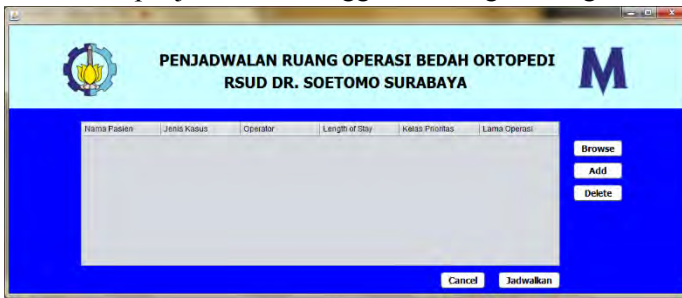
Dalam antarmuka home terdapat *button* “Get Started” yang digunakan untuk membuka jendela penjadwalan.



**Gambar 8** Antarmuka data pasien yang akan dijadwalkan

Dalam antarmuka data pasien yang akan dijadwalkan terdapat 5 *button* yaitu “Data Percobaan 1”, “Data Percobaan 2”, “Data Percobaan 3”, “Data Baru”, dan “Jadwalkan”. *Button* “Data Percobaan 1”, “Data Percobaan 2”, dan “Data Percobaan 3” digunakan untuk menampilkan data pasien yang akan dijadwalkan yang sudah disimpan sebelumnya. Data akan ditampilkan dalam tabel data pasien. Data yang ditampilkan meliputi nama pasien, jenis kasus, operator, *length of stay*, kelas prioritas, dan perkiraan lama operasi. Dari ketiga *button* ini yang membedakan antara 1 *button* dengan *button* yang lain

adalah jumlah data yang akan ditampilkan atau data yang akan dijadwalkan. Untuk *button* “Data Percobaan 1” berisi 26 data pasien. Untuk *button* “Data Percobaan 2” berisi 39 data pasien. Untuk *button* “Data Percobaan 3” berisi 49 data pasien. *Button* “Data Baru” digunakan untuk membuka antarmuka untuk menginputkan data pasien baru. *Button* “Jadwalkan” digunakan untuk membuka antarmuka penjadwalan menggunakan algoritma genetika.



**Gambar 9** Antarmuka memasukkan data pasien baru

Pada antarmuka untuk memasukkan data pasien baru terdapat 4 *button* yaitu “Add”, “Delete”, “Cancel”, dan “Jadwalkan”. *Button* “Add” digunakan untuk menambahkan baris pada tabel untuk memasukkan data pasien baru. *Button* “Delete” digunakan untuk menghapus data pada tabel di baris terakhir. *Button* “Cancel” digunakan untuk membatalkan memasukkan data pasien baru dan menutup jendela antarmuka. *Button* “Jadwalkan” digunakan untuk membuka antarmuka penjadwalan menggunakan algoritma genetika.

Antarmuka penjadwalan menggunakan algoritma genetika digunakan menjadwalkan data pasien yang diperoleh dari antarmuka sebelumnya. Dalam antarmuka ini terdapat beberapa masukan diantaranya ukuran populasi, peluang *crossover*, dan peluang mutasi. 3 hal ini merupakan parameter dari algoritma genetika. 3 parameter ini dibutuhkan untuk melakukan penjadwalan. Selain itu di panel “Parameter Algoritma Genetika” ini terdapat 3 *button* yaitu “Mulai dengan Parameter Masukan”, “Mulai Tanpa Parameter Masukan”, dan “Reset”. *Button* “Mulai dengan Parameter Masukan” digunakan untuk memulai algoritma genetika untuk melakukan penjadwalan dengan nilai parameter diperoleh dari masukan *user*. *Button* “Mulai Tanpa Parameter Masukan” digunakan untuk memulai algoritma genetika untuk melakukan penjadwalan dengan nilai parameter *default*. Jika proses penjadwalan ini selesai, maka *field* yang ada pada panel “Kendala, Fitness, dan Running Time” akan terisi sesuai dengan hasil penjadwalan. Sedangkan untuk hasil penjadwalan sendiri

akan ditampilkan dalam 3 tabel yang ada di panel “Hasil Penjadwalan” yang dibedakan berdasarkan ruang operasi. *Button* “Reset” digunakan untuk mengosongkan semua *field* dan tabel yang ada pada antarmuka.

Kemudian untuk panel dibawahnya yaitu panel “Kendala, Fitness, dan Running Time” digunakan untuk menampilkan hasil penjadwalan yang dilakukan telah memenuhi kendala yang ada atau belum, *fitness* yang dihasilkan dari penjadwalan berapa, dan berapakah lama proses yang diperlukan untuk menjalankan algoritma genetika. Hasil ini ditampilkan dalam *field* yang telah tersedia. *Field* ini akan terisi jika proses algoritma genetika telah selesai.

Sedangkan untuk panel “Hasil Penjadwalan” digunakan untuk menampilkan hasil penjadwalan yang dilakukan berdasarkan ruang operasi yang digunakan. Tabel pada panel ini menampilkan data yang meliputi nomor, hari operasi, nama pasien, operator, dan lama waktu operasi yang dibutuhkan. Untuk mencetak maupun menyimpan file hasil penjadwalan masing-masing ruang dapat menggunakan *button* “Print” yang terletak dibawah masing-masing tabel yang datanya ingin dicetak. Data pada hasil cetak tabel tersebut lebih lengkap dibandingkan dengan data pada tabel. Selain data yang ditampilkan di tabel, data pada hasil cetak tabel ditambahkan dengan data jenis kasus, *length of stay*, dan kelas prioritas. Salah satu tampilan hasil cetak penjadwalan penggunaan ruang operasi untuk ruang operasi 610 dapat dilihat pada gambar berikut ini:

<h2>Jadwal Penggunaan Ruang Operasi</h2> <h3>RSUD Dr. Soetomo Divisi Ortopedi (Ruang 610)</h3>						
Hari	Nama Pasien	Jenis Kasus	Operator	Length of Stay (Hari)	Kelas Prioritas	Lama Operasi (Jam)
Senin	Tn Mubarak	Debridement,Sural Flap.	MOY	28	Kelas 3	2.0
Senin	An Ahmad Sulaiman	Above Knee Amputasi	MED	9	Kelas 2	3.0
Selasa	Tn M Said	Amputasi Setinggi Bahu	MED	28	Kelas 2	6.0
Rabu	An Isnaina S.	Amputasi Above Knee	MED	27	Kelas Utama	3.0
Rabu	Farady B.	Skin Graft + Debridement	MED	26	Kelas 3	2.0
Kamis	Chalmatus S.	Arthroscopy Shoulder	EDY	26	Kelas Utama	3.0
Kamis	M.Ai Firmansjah	Revisi TENS Cruris (D)	SBS	24	Kelas 3	3.0
Jumat	I Made Dwi K.	FFMT C5 Th.1	HSG	20	Kelas Utama	6.0

Surabaya, 12 July 2016

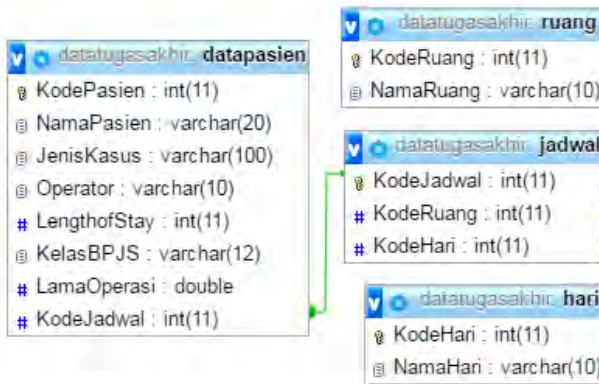
**Penanggungjawab**

**Izza**

**Gambar 11** Contoh salah satu hasil cetak jadwal untuk ruang 610

#### 4.6.2 Implementasi *Database*

Implementasi tabel dan relasi pada *database* MySQL dapat dilihat pada gambar berikut ini:



**Gambar 12** Rancangan tabel dan relasinya dalam DBMS MySQL



### 4.6.3 Implementasi Pembentukan Kromosom

Pembentukan kromosom permutasi dalam sistem penjadwalan ini diimplementasikan pada *method* `generateIndividu()`.

```
public void generateIndividu(){
    boolean[] sudahada = new
    boolean[sizeIndividu()];
    int i = 0;
    int j = 0;
    for(i=0; i<kromosom.length; i++){
        sudahada[i] = false;
    }
    for(i=0; i<kromosom.length; i++){
        int temp = (int)
        (Math.random()*kromosom.length);
        if(sudahada[temp]==false){
            kromosom[i] = temp;
            sudahada[temp] = true;
        }
        else
            i--;
    }
}
```

**Source Code 1** *Method* untuk membentuk kromosom/individu

*Method* `generateIndividu()` diatas digunakan untuk membentuk 1 kromosom elemen permutasi dalam 1 populasi.

### 4.6.4 Implementasi Seleksi

Metode seleksi kromosom dalam sistem penjadwalan ini diimplementasikan pada *method* `seleksi()` dengan parameter populasi. *Method* ini akan menghasilkan kromosom yang akan dikenai operator *crossover*.

```

private static Individu seleksi(Populasi
pop){
    Populasi tournament = new Populasi(5,
false);
    for(int i=0; i<5; i++){
        int randomId = (int)
(Math.random()*pop.size());
        tournament.saveIndividu(i,
pop.getIndividu(randomId));
    }
    Individu fittest = tournament.getFittest();

    return fittest;
}

```

**Source Code 2** *Method* untuk melakukan seleksi

#### 4.6.5 Implementasi *Crossover*

Metode *crossover* dalam sistem penjadwalan ini diimplementasikan pada *method* crossover() dengan parameter 2 kromosom yang melakukan *crossover* dan menghasilkan 2 kromosom baru.

```

private static Populasi crossover(Individu
indiv1, Individu indiv2){
    Populasi anak = new Populasi(2,false);
    Individu anak1 = new Individu();
    Individu anak2 = new Individu();
    if(Math.random()<=peluangCrossover){
        int i;
        boolean[] sudahada1 = new
boolean[indiv1.sizeIndividu()];
        boolean[] sudahada2 = new
boolean[indiv1.sizeIndividu()];
        int crosspoint =
(int)(Math.random()*indiv1.sizeIndividu());
        int indeks1=0;
        int indeks2=0;
    }
}

```

**Source Code 3** *Method* untuk melakukan *crossover*

```

for(i=0; i<sudahada1.length; i++){
    sudahada1[i] = false;
    sudahada2[i] = false;
}
for(i=0; i<crosspoint; i++){
    int val1 = indiv1.getKromosom(i);
    anak1.setKromosom(indeks1, val1);
    sudahada1[val1] = true;
    indeks1++;
    int val2 = indiv2.getKromosom(i);
    anak2.setKromosom(indeks2, val2);
    sudahada2[val2] = true;
    indeks2++;
}
for(i=0; i<crosspoint; i++){
    int val1 = indiv2.getKromosom(i);
    if(sudahada1[val1]==false){
        anak1.setKromosom(indeks1, val1);
        sudahada1[val1] = true;
        indeks1++;
    }
    int val2 = indiv1.getKromosom(i);
    if(sudahada2[val2]==false){
        anak2.setKromosom(indeks2, val2);
        sudahada2[val2] = true;
        indeks2++;}}

if(indeks1<anak1.sizeIndividu()){
    for(i=crosspoint; i<anak1.sizeIndividu();
    i++){
        int val = indiv1.getKromosom(i);
        if(sudahada1[val]==false){
            anak1.setKromosom(indeks1, val);
            sudahada1[val] = true;
            indeks1++;
        }}
    if(indeks2<anak2.sizeIndividu()){
        for(i=crosspoint; i<anak2.sizeIndividu();
        i++){
            int val = indiv2.getKromosom(i);
            if(sudahada2[val]==false){
                anak2.setKromosom(indeks2, val);

```

**Source Code 4** Lanjutan *method* untuk melakukan *crossover*

```

sudahada2[val] = true;
indeks2++;
}}
if(indeks1<anak1.sizeIndividu()){
for(i=crosspoint; i<anak1.sizeIndividu();
i++){
int val = indiv2.getKromosom(i);
if(sudahada1[val]==false){
anak1.setKromosom(indeks1, val);
sudahada1[val] = true;
indeks1++;
}}}

if(indeks2<anak2.sizeIndividu()){
for(i=crosspoint; i<anak2.sizeIndividu();
i++){
int val = indiv1.getKromosom(i);
if(sudahada2[val]==false){
anak2.setKromosom(indeks2, val);
sudahada2[val] = true;
indeks2++;
}}}}
anak.saveIndividu(0, anak1);
anak.saveIndividu(1, anak2);
return anak;
}

```

**Source Code 5** Lanjutan *method* untuk melakukan *crossover*

#### 4.6.6 Implementasi Mutasi

Metode mutasi dalam sistem penjadwalan ini diimplementasikan pada *method* mutasi() dengan parameter kromosom yang akan dimutasi.

```

private static void mutasi(Individu indiv){
    if(Math.random()<=mutasiRate){
        int bil1 = (int)
        (Math.random()*indiv.sizeIndividu());
        int bil2 = (int)
        (Math.random()*indiv.sizeIndividu());

        while(bil2==bil1){
            bil2 = (int)
            (Math.random()*indiv.sizeIndividu());}

        int temp = indiv.getKromosom(bil1);
        indiv.setKromosom(bil1,
        indiv.getKromosom(bil2));
        indiv.setKromosom(bil2, temp);
    }}

```

**Source Code 6** *Method* untuk melakukan mutasi

#### 4.6.7 Implementasi Fungsi *Fitness*

Evaluasi perhitungan nilai *fitness* dari suatu kromosom diimplementasikan pada *method* `getFitness()` dengan parameter kromosom yang akan dihitung nilai *fitness*-nya. Keluaran dari *method* ini yaitu nilai *fitness* dari kromosom yang dijadikan sebagai parameter saat memanggil *method* tersebut.

```

static double getFitness(Individu individu){
    double Fitness=0;
    double urgent = cekUrgensitas(individu);
    double operator = cekOperator(individu);
    double length = cekLengthofStay(individu);
    double bpjs = cekBPJS(individu);
    double h = getFit(individu)[0];
    double r = getFit(individu)[1];
    Fitness =
    ((urgent*1000000)+(operator*100000)+(length*
    10000)+(bpjs*1000)+(h*10)+r);
    return Fitness;
}

```

**Source Code 7** *Method* untuk fungsi *fitness*

#### 4.6.8 Implementasi *Elitism*

Proses *elitism* diimplementasikan pada *method* *elitism()* dengan 2 parameter populasi yang akan disaring. Proses ini menghasilkan populasi baru dengan jumlah kromosom sama dengan ukuran populasi.

```
public Populasi elitism(Populasi pop1,
Populasi pop2){
Populasi newPopulation = new
Populasi(pop1.size(), false);
Populasi populasi = new
Populasi(pop1.size()+pop2.size(), false);
double fitness[] = new
double[pop1.size()+pop2.size()];

for(int i=0; i<pop1.size(); i++){
populasi.saveIndividu(i,
pop1.getIndividu(i));
}
for(int i=0; i<pop2.size(); i++){
populasi.saveIndividu(i+pop1.size(),
pop2.getIndividu(i));
}

for(int i=0; i<populasi.size(); i++){
fitness[i] =
Fitness.getFitness (populasi.getIndividu(i));
}

for(int i=0; i<populasi.size(); i++){
for(int j=0; j<populasi.size()-1; j++){
if(fitness[j+1]<fitness[j]){
Individu temp = populasi.getIndividu(j+1);
populasi.saveIndividu(j+1,
populasi.getIndividu(j));
populasi.saveIndividu(j, temp);
double temp1 = fitness[j+1];
fitness[j+1] = fitness[j];
fitness[j] = temp1;
}}}
```

**Source Code 8** *Method* untuk melakukan proses *elitism*

```

int index = 0;
newPopulation.saveIndividu(0,
populasi.getIndividu(0));
for(int i=1; i<populasi.size()-popl.size();
i++){
if(fitness[i]!=fitness[index]){
index++;
newPopulation.saveIndividu(index,
populasi.getIndividu(i));
}
}
if(index!=popl.size()-1){
for(int i=index+1; i<popl.size(); i++){
Individu newIndividu = new Individu();
newIndividu.generateIndividu();
newPopulation.saveIndividu(i, newIndividu);
}
}

return newPopulation;
}

```

**Source Code 9** Lanjutan *method* untuk melakukan proses *elitism*





## **BAB V**

### **PENGUJIAN DAN PEMBAHASAN HASIL**

Pada bab ini dijelaskan mengenai pengujian yang dilakukan terhadap masalah penjadwalan penggunaan ruang operasi dengan menggunakan metode algoritma genetika. Hasil pengujian kemudian dibahas untuk mengetahui kinerja sistem secara keseluruhan dalam menjalankan fungsi yang diharapkan.

#### **5.1 Pengujian Pengaruh Kombinasi Parameter Terhadap Kestabilan Nilai *Fitness***

Pengujian pengaruh peluang *crossover* dan mutasi pada sistem ini dilakukan untuk menguji data sebanyak 26 pasien dengan 36 kombinasi parameter algoritma genetika yaitu ukuran populasi, peluang *crossover*, dan peluang mutasi. Dengan ukuran populasi 30, 50, dan 80. Sedangkan peluang *crossover* sebesar 1, 0.8 dan 0.5 serta peluang mutasi sebesar 0.01, 0.1, 0.3, dan 0.5. Jumlah generasi maksimum akan ditentukan oleh sistem. Iterasi berhenti jika nilai *fittest* yang diperoleh sudah memenuhi semua kendala dan tidak berubah selama 2000 iterasi atau generasi telah mencapai 100000 generasi. Masing-masing kombinasi diuji sebanyak 5 kali.

Dalam pengujian diperoleh bahwa nilai *fitness* dominan yang muncul adalah 53. Berdasarkan pengujian tersebut, banyaknya nilai dominan muncul untuk masing-masing kombinasi parameter algoritma genetika ditunjukkan oleh tabel berikut ini:

**Tabel 2** Hasil pengujian banyaknya nilai *fitness* dominan yang muncul

<b>Ukuran Populasi</b>	<b>Peluang Crossover</b>	<b>Peluang Mutasi</b>	<b>Jumlah Kemunculan</b>
30	1	0.01	4 kali
30	1	0.1	4 kali
30	1	0.3	4 kali
<b>30</b>	<b>1</b>	<b>0.5</b>	<b>5 kali</b>
30	0.8	0.01	4 kali
30	0.8	0.1	2 kali
<b>30</b>	<b>0.8</b>	<b>0.3</b>	<b>5 kali</b>
<b>30</b>	<b>0.8</b>	<b>0.5</b>	<b>5 kali</b>
30	0.5	0.01	4 kali
30	0.5	0.1	3 kali
30	0.5	0.3	3 kali
30	0.5	0.5	4 kali
50	1	0.01	4 kali
<b>50</b>	<b>1</b>	<b>0.1</b>	<b>5 kali</b>
<b>50</b>	<b>1</b>	<b>0.3</b>	<b>5 kali</b>
50	1	0.5	4 kali
50	0.8	0.01	4 kali
<b>50</b>	<b>0.8</b>	<b>0.1</b>	<b>5 kali</b>
50	0.8	0.3	4 kali
<b>50</b>	<b>0.8</b>	<b>0.5</b>	<b>5 kali</b>
50	0.5	0.01	3 kali
<b>50</b>	<b>0.5</b>	<b>0.1</b>	<b>5 kali</b>
50	0.5	0.3	4 kali
<b>50</b>	<b>0.5</b>	<b>0.5</b>	<b>5 kali</b>
80	1	0.01	4 kali
<b>80</b>	<b>1</b>	<b>0.1</b>	<b>5 kali</b>
<b>80</b>	<b>1</b>	<b>0.3</b>	<b>5 kali</b>
<b>80</b>	<b>1</b>	<b>0.5</b>	<b>5 kali</b>
80	0.8	0.01	4 kali

80	0.8	0.1	4 kali
<b>80</b>	<b>0.8</b>	<b>0.3</b>	<b>5 kali</b>
<b>80</b>	<b>0.8</b>	<b>0.5</b>	<b>5 kali</b>
80	0.5	0.01	4 kali
80	0.5	0.1	3 kali
80	0.5	0.3	4 kali
80	0.5	0.5	4 kali

Dari data diatas terlihat bahwa hasil dari kombinasi parameter diatas tidak terlalu mempengaruhi nilai *fitness* yang dihasilkan. Hal ini disebabkan oleh kriteria pemberhentian yang telah didefinisikan. Dari data diatas terdapat 14 kombinasi yang menghasilkan 5 kali muncul nilai *fitness* dominan. Kombinasi dengan ukuran populasi 30 yang nilai *fitness* dominannya muncul sebanyak 5 kali hanya sebanyak 3 kombinasi saja. Sedangkan untuk kombinasi dengan ukuran populasi 50 dan 80 yang nilai *fitness* dominannya muncul sebanyak 5 kali terdapat 6 dan 5 kombinasi secara berturut-turut. Nilai peluang *crossover* yang selalu menghasilkan nilai *fitness* dominan sebanyak 3 yaitu 1, 0.8 dan 0.5. Namun, untuk nilai peluang *crossover* 0.5 hanya muncul 2 kali saja dengan ukuran populasinya 50. Untuk nilai peluang mutasi sebesar 0.01 tidak ada yang selalu menghasilkan nilai *fitness* dominan. Hal ini menunjukkan bahwa nilai peluang mutasi untuk permasalahan penjadwalan ruang operasi tidak boleh terlalu kecil.

Sehingga berdasarkan data tersebut dapat diperoleh kesimpulan bahwa ukuran populasi yang baik adalah 50, dengan peluang *crossover* sebesar 0.8 dan peluang mutasi sebesar 0.5.

## 5.2 Pengujian Pengaruh Peluang *Crossover* dan Mutasi pada Sistem Terhadap Waktu Komputasi

Pengujian pengaruh peluang *crossover* dan mutasi terhadap waktu komputasi pada sistem ini dilakukan untuk menguji data sebanyak 26 pasien dengan 14 kombinasi percobaan. Berdasarkan pengujian sebelumnya 14 kombinasi inilah yang menghasilkan nilai *fitness* yang stabil. Pada pengujian ini akan dilihat waktu komputasi dari tiap kombinasi untuk 5 kali uji. Setelah mendapatkan waktu komputasi per kombinasinya dicari rata-rata waktu komputasi dari masing-masing kombinasi. Rata-rata waktu komputasi ini yang menunjukkan seberapa cepat kombinasi tersebut dapat menghasilkan nilai *fitness* yang stabil. Hasil pengujian ditunjukkan pada Tabel 2.

Dari Tabel 2 terlihat bahwa rata-rata waktu komputasi yang dibutuhkan dalam setiap kombinasi cukup beragam. Kombinasi dengan ukuran populasi 80 memiliki waktu komputasi yang cukup lama dibandingkan dengan kombinasi lainnya. Rata-rata waktu komputasi terkecil yaitu untuk kombinasi dengan ukuran populasi 30, peluang *crossover* 1 dan peluang mutasi sebesar 0.5.

Dari pengujian sebelumnya diperoleh informasi bahwa kombinasi terbaik adalah ukuran populasi sama dengan 50, dengan peluang *crossover* sebesar 0.8 dan peluang mutasi sebesar 0.5. Kemudian berdasarkan informasi rata-rata waktu komputasinya pun tidak terlalu lama. Sehingga kombinasi inilah yang baik untuk digunakan dalam permasalahan ini.

**Tabel 3** Hasil pengujian pengaruh peluang crossover dan mutasi pada sistem terhadap waktu komputasi

Ukuran Populasi	Peluang Crossover	Peluang Mutasi	Waktu Komputasi (detik) Pengujian ke					Rata-Rata
			1	2	3	4	5	
30	1	0.5	259	207	452	291	140	269,8
30	0.8	0.3	538	554	622	429	159	460,4
30	0.8	0.5	298	559	293	246	151	309,4
50	1	0.1	548	641	512	1266	1702	933,8
50	1	0.3	650	374	537	382	355	459,6
50	0.8	0.1	356	230	423	450	323	356,4
50	0.8	0.5	444	287	204	435	311	336,2
50	0.5	0.1	307	344	412	365	606	406,8
50	0.5	0.5	235	365	391	362	388	348,2
80	1	0.1	1576	1058	1222	1319	1312	1297,4
80	1	0.3	1181	1445	1447	1567	1603	1448,6
80	1	0.5	1428	1766	1559	1688	1763	1640,8
80	0.8	0.3	1568	1455	1616	2485	3157	2056,2
80	0.8	0.5	1473	705	711	800	1062	950,2

### 5.3 Pengujian Banyaknya Pasien yang Dijadwalkan

Berdasarkan pengujian sebelumnya dapat dilihat bahwa kombinasi yang menghasilkan nilai *fitness* yang stabil dengan waktu komputasi cukup cepat yaitu kombinasi dengan ukuran populasi 50, peluang *crossover* 0.8, dan peluang mutasi 0.5. Sehingga, dalam pengujian banyaknya pasien yang dijadwalkan oleh sistem digunakan parameter tersebut. Untuk data diatas 26 dibuat kriteria pemberhentian yaitu berhenti jika nilai *fittest* yang diperoleh sudah memenuhi semua kendala dan tidak berubah selama 2500 iterasi atau generasi telah mencapai 100000 generasi.

**Tabel 4** Hasil pengujian banyaknya pasien yang dijadwalkan

Jumlah Pasien	Nilai <i>Fitness</i>	Waktu Komputasi (detik)
3	12	14
5	21	50
26	53	189
39	73	840
44	83	1356
49	101	5109

Berdasarkan data diatas dapat diketahui bahwa semakin banyak jumlah pasien yang dijadwalkan semakin lama waktu komputasi yang diperlukan. Untuk nilai *fitness*-nya sampai dengan 49 pasien masih memenuhi semua kendala yang ada. Hal ini terlihat dari niali *fitness*-nya yang kurang dari 999.

## LAMPIRAN

### A. *Source Code*

#### ***Class individu***

```
package tugasakhir;
public class Individu {
    static int defaultKromosomLength = 100;
    int [] kromosom = new
    int[defaultKromosomLength];
    int fitness = 0;
    static double [] waktuOperasi = new
    double[defaultKromosomLength];
    static int [] operator = new
    int[defaultKromosomLength];
    static int [] lengthofStay = new
    int[defaultKromosomLength];
    static byte [] urgensiitas = new
    byte[defaultKromosomLength];
    static int [] kelasBPJS = new
    int[defaultKromosomLength];

    public void generateIndividu(){
        boolean[] sudahada = new
        boolean[sizeIndividu()];
        int i = 0;
        int j = 0;
        for(i=0; i<kromosom.length; i++){
            sudahada[i] = false;
        }
        for(i=0; i<kromosom.length; i++){
            int temp = (int)
            (Math.random()*kromosom.length);
            if(sudahada[temp]==false){
                kromosom[i] = temp;
                sudahada[temp] = true;
            }
            else
                i--;
        }

        public static void
        setDefaultKromosomLength(int length){
```

```

defaultKromosomLength = length;
}

public int getKromosom(int index){
return kromosom[index];
}

public void setKromosom(int index, int
value){
kromosom[index] = value;
}

public static double getWaktuOperasi(int
index){
return waktuOperasi[index];
}

public static void setWaktuOperasi(int index,
double value){
waktuOperasi[index] = value;
}

public static int getOperator(int index){
return operator[index];
}

public static void setOperator(int index, int
value){
operator[index] = value;
}

public static int getLengthofStay(int index){
return lengthofStay[index];
}

public static void setLengthofStay(int index,
int value){
lengthofStay[index] = value;
}

public static byte getUrgensitas(int index){
return urgensitas[index];}

```



```

public static void setUrgensitas(int index,
byte value){
urgensitas[index] = value;
}

public static int getKelasBPJS(int index){
return kelasBPJS[index];
}

public static void setKelasBPJS(int index,
int value){
kelasBPJS[index] = value;
}

public int sizeIndividu(){
return kromosom.length;
}

@Override
public String toString(){
String kromosomString = "";
for(int i=0; i<sizeIndividu(); i++){
kromosomString += getKromosom(i);
}
return kromosomString;
}

public double[] getKromosomFit(int kromosom){
double fitness[] = new double[3];
int hari=1; int n=0; double total=0; int i;
int ruang=0; double waktu=0; int indeks=0;
for(i=0; i<sizeIndividu(); i++){
if(getKromosom(i)==kromosom){
indeks = i;
break;
}}
while(n<=indeks){
for(i=n; i<=indeks; i++){
int temp = getKromosom(i);
total = total+getWaktuOperasi(temp)+0.5;
waktu = total;
if(total>7){
n = i;
ruang++;
}
}
}
}

```

```

total = 0;
if(ruang%4==0){
hari++;
ruang = 1;
}
break;
}
else if(total==7){
n = i+1;
ruang++;
total = 0;
if(ruang%4==0){
hari++;
ruang = 1;
}
break;
}
else if(i==indeks){
n = i+1;
ruang++;
if(ruang%4==0){
hari++;
ruang = 1;
}
break;
}}
fitness[0] = hari;
fitness[1] = ruang;
fitness[2] = waktu;
return fitness;
}

public int[][] operatorSama(){
int[][] operatorSama = new
int[15][sizeIndividu()];
int a=0, b=0, c=0, d=0, e=0, f=0, g=0, h=0,
i=0, j=0, k=0, l=0, m=0, n=0, o=0;
for(int x=0; x<sizeIndividu(); x++){
for(int y=0; y<15; y++){
operatorSama[y][x] = -1;
}
for(int x=0; x<sizeIndividu(); x++){
if(getOperator(x)==0){
operatorSama[0][a] = x;

```

```
a++;  
}  
else if(getOperator(x)==1){  
operatorSama[1][b] = x;  
b++;  
}  
else if(getOperator(x)==2){  
operatorSama[2][c] = x;  
c++;  
}  
else if(getOperator(x)==3){  
operatorSama[3][d] = x;  
d++;  
}  
else if(getOperator(x)==4){  
operatorSama[4][e] = x;  
e++;  
}  
else if(getOperator(x)==5){  
operatorSama[5][f] = x;  
f++;  
}  
else if(getOperator(x)==6){  
operatorSama[6][g] = x;  
g++;  
}  
else if(getOperator(x)==7){  
operatorSama[7][h] = x;  
h++;  
}  
else if(getOperator(x)==8){  
operatorSama[8][i] = x;  
i++;  
}  
else if(getOperator(x)==9){  
operatorSama[9][j] = x;  
j++;  
}  
else if(getOperator(x)==10){  
operatorSama[10][k] = x;  
k++;  
}  
else if(getOperator(x)==11){  
operatorSama[11][l] = x;
```

```

l++;
}
else if(getOperator(x)==12){
operatorSama[12][m] = x;
m++;
}
else if(getOperator(x)==13){
operatorSama[13][n] = x;
n++;
}
else if(getOperator(x)==14){
operatorSama[14][o] = x;
o++;
}}
return operatorSama;
}

public int[][] UrgensitasSama(){
int[][] urgent = new int[2][sizeIndividu()];
int indeks = 0;
int indeks1 = 0;
for(int i=0; i<2; i++){
for(int j=0; j<sizeIndividu(); j++)
urgent[i][j] = -1;
}
for(int i=0; i<sizeIndividu(); i++){
if(getUrgensitas(getKromosom(i))==0){
urgent[0][indeks] = i;
indeks++;
}
else{
urgent[1][indeks1] = i;
indeks1++;
}}
return urgent;
}

public int[][] KelasBPJSSama(){
int[][] bpjs = new int[4][sizeIndividu()];
int indeks = 0;
int indeks1 = 0;
int indeks2 = 0;
int indeks3 = 0;

```

```

for(int i=0; i<4; i++){
for(int j=0; j<sizeIndividu(); j++)
bpjs[i][j] = -1;
}
for(int i=0; i<sizeIndividu(); i++){
if(getKelasBPJS(getKromosom(i))==0){
bpjs[0][indeks] = i;
indeks++;
}
else if(getKelasBPJS(getKromosom(i))==1){
bpjs[1][indeks1] = i;
indeks1++;
}
else if(getKelasBPJS(getKromosom(i))==2){
bpjs[2][indeks2] = i;
indeks2++;
}
else if(getKelasBPJS(getKromosom(i))==3){
bpjs[3][indeks2] = i;
indeks3++;
}}
return bpjs;
}}

```

### ***Class populasi***

```

package tugasakhir;
public class Populasi {
Individu[] individu;

public Populasi(int populasiSize, boolean
initialise){
individu = new Individu[populasiSize];
if(initialise){
for(int i=0; i<size(); i++){
Individu newIndividu = new Individu();
newIndividu.generateIndividu();
saveIndividu(i, newIndividu);
}}}

public Individu getIndividu(int index){
return individu[index];
}
}

```

```

public Individu getFittest(){
Individu fittest = individu[0];
for(int i=1; i<size(); i++){
if(Fitness.getFitness(fittest)>Fitness.getFitness(getIndividu(i))){
fittest = getIndividu(i);
}}
return fittest;
}

public int size(){
return individu.length;
}

public void saveIndividu(int index, Individu
indiv){
individu[index] = indiv;
}

public Populasi elitism(Populasi pop1,
Populasi pop2){
Populasi newPopulation = new
Populasi(pop1.size(), false);
Populasi populasi = new
Populasi(pop1.size()+pop2.size(), false);
double fitness[] = new
double[pop1.size()+pop2.size()];

for(int i=0; i<pop1.size(); i++){
populasi.saveIndividu(i,
pop1.getIndividu(i));
}
for(int i=0; i<pop2.size(); i++){
populasi.saveIndividu(i+pop1.size(),
pop2.getIndividu(i));
}
for(int i=0; i<populasi.size(); i++){
fitness[i] =
Fitness.getFitness(populasi.getIndividu(i));
}

for(int i=0; i<populasi.size(); i++){
for(int j=0; j<populasi.size()-1; j++){
if(fitness[j+1]<fitness[j]){

```

```

Individu temp = populasi.getIndividu(j+1);
populasi.saveIndividu(j+1,
populasi.getIndividu(j));
populasi.saveIndividu(j, temp);
double temp1 = fitness[j+1];
fitness[j+1] = fitness[j];
fitness[j] = temp1;
}}

int index = 0;
newPopulation.saveIndividu(0,
populasi.getIndividu(0));
for(int i=1; i<populasi.size()-pop1.size();
i++){
if(fitness[i]!=fitness[index]){
index++;
newPopulation.saveIndividu(index,
populasi.getIndividu(i));
}}

if(index!=pop1.size()-1){
for(int i=index+1; i<pop1.size(); i++){
Individu newIndividu = new Individu();
newIndividu.generateIndividu();
newPopulation.saveIndividu(i, newIndividu);
}}

return newPopulation;
}}

```

### ***Class fitness***

```

package tugasakhir;
public class Fitness {
static double[] getFit(Individu individu){
double fitness[] = new double[3];
int hari=1; int n=0; double total=0; int i;
int ruang=0; double waktu=0;

while(n<individu.sizeIndividu()){
for(i=n; i<individu.sizeIndividu(); i++){
int temp = individu.getKromosom(i);
total =
total+individu.getWaktuOperasi(temp)+0.5;

```

```

    waktu = total;
    if(total>7){
        n = i;
        ruang++;
        total = 0;
        if(ruang%4==0){
            hari++;
            ruang = 1;
        }
        break;
    }
    else if(total==7){
        n = i+1;
        ruang++;
        total = 0;
        if(ruang%4==0){
            hari++;
            ruang = 1;
        }
        break;
    }
    else if(i==individu.sizeIndividu()-1){
        n = i+1;
        ruang++;
        if(ruang%4==0){
            hari++;
            ruang = 1;
        }
        break;
    }
    fitness[0] = hari;
    fitness[1] = ruang;
    fitness[2] = waktu;
    return fitness;
}

static double getFitness(Individu individu){
    double Fitness=0;
    double urgent = cekUrgensitas(individu);
    double operator = cekOperator(individu);
    double length = cekLengthofStay(individu);
    double bpjs = cekBPJS(individu);
    double h = getFit(individu)[0];
    double r = getFit(individu)[1];

```



```

Fitness =
((urgent*1000000)+(operator*100000)+(length*1
0000)+(bpjs*1000)+(h*10)+r);
return Fitness;
}

static double getMinFitness(Populasi
populasi){
double minFitness =
getFitness(populasi.getFittest());
return minFitness;
}

static double cekUrgensitas(Individu
individu){
boolean salah = false;
double cek = 0;
double u;
for(int i=0; i<individu.sizeIndividu(); i++){
if(individu.getUrgensitas(individu.getKromoso
m(i))==1){
if(individu.getKromosomFit(individu.getKromoso
m(i))[0]>5){
u =
individu.getKromosomFit(individu.getKromosom(
i))[0]+individu.getLengthofStay(individu.getK
romosom(i))+Math.ceil(individu.getKromosomFit
(individu.getKromosom(i))[0]/5.0);
}
else{
u =
individu.getKromosomFit(individu.getKromosom(
i))[0]+individu.getLengthofStay(individu.getK
romosom(i));
}
if(u>=14)
cek++;
}}
return cek;
}

static double cekBPJS(Individu individu){
double cek = 0;

```

```

int[][] urgent = new
int[2][individu.sizeIndividu()];
urgent = individu.UrgensitasSama();
for(int a=0; a<2; a++){
for(int i=0; i<individu.sizeIndividu(); i++){
if(urgent[a][i]!=-1){
for(int j=i+1; j<individu.sizeIndividu();
j++){
if(urgent[a][j]!=-1 &&
individu.getOperator(individu.getKromosom(urg
ent[a][i]))==individu.getOperator(individu.ge
tKromosom(urgent[a][j]))
&&
individu.getLengthofStay(individu.getKromosom
(urgent[a][i]))==individu.getLengthofStay(ind
ividu.getKromosom(urgent[a][j]))){
double x =
individu.getKromosomFit(individu.getKromosom(
i))[0];
double y =
individu.getKromosomFit(individu.getKromosom(
j))[0];
if(individu.getKelasBPJS(individu.getKromosom
(urgent[a][i]))>individu.getKelasBPJS(individ
u.getKromosom(urgent[a][j])) && x>=y){
cek++;
}}}}}}
return cek;
}

```

```

static double cekLengthofStay(Individu
individu){
double cek = 0;
double z;
int[][] urgent = new
int[2][individu.sizeIndividu()];
urgent = individu.UrgensitasSama();
for(int a=0; a<2; a++){
for(int i=0; i<individu.sizeIndividu(); i++){
if(urgent[a][i]!=-1){
for(int j=i+1; j<individu.sizeIndividu();
j++){

```

```

double x =
individu.getKromosomFit(individu.getKromosom(
urgent[a][i]))[0];
if(x>5){
z = x+(Math.ceil(x/5.0));
}
else{
z = x;
}
if(urgent[a][j]!=-1 &&
individu.getOperator(individu.getKromosom(urg
ent[a][i]))==individu.getOperator(individu.ge
tKromosom(urgent[a][j]))){
if(individu.getLengthofStay(individu.getKromo
som(urgent[a][i]))<individu.getLengthofStay(i
ndividu.getKromosom(urgent[a][j]))){
cek++;
}}
else
if(individu.getUrgensitas(individu.getKromoso
m(urgent[a][i]))==0 &&
(z+individu.getLengthofStay(individu.getKromo
som(urgent[a][i]))<14 ||
z+individu.getLengthofStay(individu.getKromos
om(urgent[a][i]))>30)){
cek++;
}}}}
return cek;
}

static double cekOperator(Individu individu){
int[][] operatorSama = new
int[15][individu.sizeIndividu()];
double cek = 0;
operatorSama = individu.operatorSama();
for(int x=0; x<operatorSama.length; x++){
for(int y=0; y<individu.sizeIndividu(); y++){
if(operatorSama[x][y]!=-1){
int temp = operatorSama[x][y];
for(int z=y+1; z<individu.sizeIndividu();
z++){
if(operatorSama[x][z]!=-1){
int op = operatorSama[x][z];

```

```

if (individu.getKromosomFit(temp)[0]==individu
.getKromosomFit(op)[0] &&
individu.getKromosomFit(temp)[1]!=individu.ge
tKromosomFit(op)[1]){
if (individu.getKromosomFit(temp)[2]<individu.
getKromosomFit(op)[2]){
double t = individu.getKromosomFit(op)[2]-
individu.getWaktuOperasi(op);
if (t<individu.getWaktuOperasi(temp))
cek++;
}
else{
double t = individu.getKromosomFit(temp)[2]-
individu.getWaktuOperasi(temp);
if (t<individu.getWaktuOperasi(op))
cek++;
}}}}}
else
break;
}}
return cek;
}}

```

## B. Hasil Penjadwalan

Hasil penjadwalan untuk 3 pasien

Jadwal Penggunaan Ruang Operasi RSUD Dr. Soetomo Divisi Ortopedi (Ruang 610)						
No.	Nama Pasien	Jenis Kelamin	Operator	Length of Stay (Hari)	Kelas Rawat	Lama Operasi (Jam)
01	Tu M. Sidi	L	Arif Nugroho Satrio Nugroho	28	Kelas I	00:00

Jadwal Penggunaan Ruang Operasi RSUD Dr. Soetomo Divisi Ortopedi (Ruang 612)						
No	Nama Pasien	Jenis Kasus	Operator	Length of Stay (Hari)	Bed Occupancy	Lama Operasi (Jam)
001	Indira Prati	Removal Hipart	JR	28	Kamar 1	3.0
002	Tn. Dikdik	Distrofisme Sendi Tangan	MDY	29	Kamar 1	2.0

Hasil penjadwalan untuk 5 pasien

Jadwal Penggunaan Ruang Operasi						
RSUD Dr. Soetomo Divisi Ortopedi (Ruang 610)						
Hari	Nama Pasien	Jenis Kasus	Operator	Length of Stay (Hari)	Kelas Prioritas	Lama Operasi (Jam)
Senin	Tr.Misnan	AF Plst Metatarsal dig.1	PPDS	28	Kelas 2	6.0
Selasa	Tr.M.Said	Amputasi Selinggi Bahu	MED	28	Kelas 2	6.0
Surabaya, 14 July 2016						
Penanggung Jawab						
Izza						

Jadwal Penggunaan Ruang Operasi						
RSUD Dr. Soetomo Divisi Ortopedi (Ruang 612)						
Hari	Nama Pasien	Jenis Kasus	Operator	Length of Stay (Hari)	Kelas Prioritas	Lama Operasi (Jam)
Senin	Ny.Ngalining	Plating Acetabulum,	PPDS	27	Kelas 3	6.0
Surabaya, 14 July 2016						
Penanggung Jawab						
Izza						

Thursday 14 July	Page 1 of 1
------------------	-------------

### Jadwal Penggunaan Ruang Operasi RSUD Dr. Soetomo Divisi Ortopedi (Ruang 614)

Hari	Nama Pasien	Jenis Kasus	Operator	Length of Stay (Hari)	Kelas Prioritas	Lama Operasi (Jam)
Senin	Andika Roza	Removal Ilizarof	JUP	28	Kelas Utama	3.0
Senin	Th. Mubarak	Debridement, Sural Flap	MOY	28	Kelas 3	2.0

Surabaya, 14 July 2016  
Penanggungjawab

Izza

Thursday 14 July

Page 1 of 1

### Hasil penjadwalan untuk 26 pasien

### Jadwal Penggunaan Ruang Operasi RSUD Dr. Soetomo Divisi Ortopedi (Ruang 610)

Hari	Nama Pasien	Jenis Kasus	Operator	Length of Stay (Hari)	Kelas Prioritas	Lama Operasi (Jam)
Senin	Th. Misnan	AF Plat Metatarsal dig. I	PPDS	28	Kelas 2	6.0
Selasa	Ny. Ngainining	Plating Acetabulum	PPDS	27	Kelas 3	6.0
Rabu	Th. Sugho	Anterior Cervical	Prof. BP	27	Kelas 2	6.0
Kamis	Th. Tulliman	AF Ext. Fixasi	PPDS	25	Kelas 2	2.0
Kamis	Th. Darmawan	PSR	PPDS	24	Kelas 2	1.0
Kamis	Emawati	Plating Tibia Plate	PPDS	23	Kelas 3	2.0
Jumat	Deddy Gunawan	Tibial Posterior Tendon	EDY	24	Kelas Utama	2.0
Jumat	Ny. Siti Kodjah	AF Plate Fibula	PPDS	22	Kelas 3	3.0

Surabaya, 14 July 2016  
Penanggungjawab

Izza

Thursday 14 July

Page 1 of 2

Jadwal Penggunaan Ruang Operasi RSUD Dr. Soetomo Divisi Ortopedi (Ruang 612)						
Hari	Nama Pasien	Jenis Kasus	Operator	Length of Stay (Hari)	Kelas Prioritas	Lama Operasi (Jam)
Senin	Tn.M.Said	Amputasi Setinggi Bahu	MED	28	Kelas 2	6.0
Selasa	Andika Reza	Removal Ilizarof	JUP	28	Kelas Utama	3.0
Selasa	An.Ahmad Sulaiman	Above Knee Amputasi	MED	9	Kelas 2	3.0
Rabu	Tn.Ahmad Yajid	Debridement + Re ORIF	EDY	27	Kelas Utama	2.0
Rabu	Shi Wajugeng	Arthroscopy +	MOY	26	Kelas Utama	2.0
Rabu	Tanso W.	Open Biopsi Multiple	BEE	20	Kelas Utama	1.0
Kamis	Lukman H.	FFMT	HSO	24	Kelas 2	6.0
Jumat	I Made Dwi K.	FFMT C5 Th.1	HSO	20	Kelas Utama	6.0
Surabaya, 14 July 2016 PenanggungJawab						
Izza						
Thursday 14 July				Page 1 of 2		

Jadwal Penggunaan Ruang Operasi RSUD Dr. Soetomo Divisi Ortopedi (Ruang 614)						
Hari	Nama Pasien	Jenis Kasus	Operator	Length of Stay (Hari)	Kelas Prioritas	Lama Operasi (Jam)
Senin	Tn.Mubarak	Debridement,Sural Flap,	MOY	28	Kelas 3	2.0
Senin	An.Ahmad Anif	Disarticulasi Hip (D)	BEE	9	Kelas 3	2.0
Selasa	An.Irwinia S.	Amputasi Above Knee	MED	27	Kelas Utama	3.0
Selasa	Farady B.	Skin Graft + Debridement	MED	26	Kelas 3	2.0
Rabu	Dewi S.	Open Biopsi	PPDS	26	Kelas 1	3.0
Rabu	Achmad N.	Serial Plastering	ARN	20	Kelas 2	2.0
Kamis	Chalimatus S.	Arthroscopy Shoulder	EDY	26	Kelas Utama	3.0
Kamis	M.Ali Firmansjah	Revisi TENS Cruris (D)	SBS	24	Kelas 3	3.0
Jumat	Tn.Siamet	AF Ext.Fixasi	MCH	24	Kelas Utama	2.0
Surabaya, 14 July 2016 PenanggungJawab						
Izza						
Thursday 14 July				Page 1 of 2		



Hari	Nama Pasien	Jenis Kasus	Operator	Length of Stay (Hari)	Kelas Prioritas	Lama Operasi (Jam)
Jumat	Ny Hana Fatmawati	Stabilisasi Posterior +	PPDS	21	Kelas 3	2.0

Surabaya, 14 July 2018  
Penanggung Jawab

Izza

---

Thursday 14 July Page 2 of 2

## Hasil penjadwalan untuk 39 pasien

<b>Jadwal Penggunaan Ruang Operasi</b> <b>RSUD Dr. Soetomo Divisi Ortopedi (Ruang 610)</b>						
Hari	Nama Pasien	Jenis Kasus	Operator	Length of Stay (Hari)	Kelas Prioritas	Lama Operasi (Jam)
Senin	Ty M Sidi	Amputasi Setinggi Kaki	ABD	28	Kelas 2	8.0
Senin	Agustina Lusi	Fraktur Humerus	ABD	25	Kelas Utama	3.0
Selasa	Ty Mubriz	Dislokasi/ Fraktur Humerus	ABD	28	Kelas 1	2.0
Rabu	Liliana H.	PPST	ABD	34	Kelas 2	8.0
Kamis	Surya Wang	Amputasi +	ABD	36	Kelas Utama	9.0
Kamis	Ty Darmawan	PPST	PPDS	34	Kelas 2	1.0
Kamis	Andang M.	Stabilisasi Posterior	ABD	35	Kelas 1	2.0
Jumat	Ny Sari Sukirni	AR Fraktur Humerus	PPST	33	Kelas 2	3.0
Jumat	Ny Hana Fatmawati	Stabilisasi Posterior +	PPDS	31	Kelas 3	2.0

Surabaya, 12 July 2018  
Penanggung Jawab

Izza

---

Thursday 12 July 2018 Page 1 of 2

Hari	Nama Pasien	Jenis Kasus	Operator	Length of Stay (Hari)	Kelas Prioritas	Lama Operasi (Jam)
Kamis	Tn.Turman	AF Ext.Fixasi	PPDS	25	Kelas 2	2.0
Kamis	Tn.Darmawan	PSR	PPDS	24	Kelas 2	1.0
Kamis	Emawati	Plating Tibia Plate	PPDS	23	Kelas 3	2.0
Jumat	Deddy Gunawan	Tibial Posterior Tendon	EDY	24	Kelas Utama	2.0
Jumat	Achmad N.	Serial Plestering	ARN	20	Kelas 2	2.0
Jumat	Tanso W.	Open Bepsi Multiple	BEE	20	Kelas Utama	1.0
Surabaya – 13 July 2019						
Penanggung Jawab						
Izza						
Wednesday 13 July				Page 2 of 2		

Jadwal Penggunaan Ruang Operasi						
RSUD Dr. Soetomo Divisi Ortopedi (Ruang 612)						
Hari	Nama Pasien	Jenis Kasus	Operator	Length of Stay (Hari)	Kelas Prioritas	Lama Operasi (Jam)
Senin	Endika Kisa	Bursural Riza of	JUP	28	Kelas Utama	3.0
Senin	Fa.Siam	AF Ext.Fixasi	MCH	24	Kelas Utama	2.0
Senin	1.1888.July.6.	PPMT C3 Tr-1	WGO	20	Kelas Utama	6.0
Selasa	Fa.Makipul	Distraction (1.5ul of Pulp)	MDY	28	Kelas 3	2.0
Selasa	Spr.Vikang	Arthroscopy +	MDY	28	Kelas Utama	2.0
Selasa	Fa.Siwach	OTOTOMAS	PRUD	18	Kelas 2	2.0
Selasa	Nori M.A.	DRP Tibia Pterial	MCH	6	Kelas Utama	2.0
Rabu	Ay.Nugling	Plating Anklestabilis	PPDS	27	Kelas 3	6.0
Surabaya – 12 July 2019						
Penanggung Jawab						
Izza						
Wednesday 13 July				Page 1 of 2		

hari	Nama Pasien	Jenis Kasus	Operator	Length of Stay (hari)	Kelas Prioritas	Lama Operasi (jam)
Kamis	Ny. Sa Kidjan	AT Pilon Fraktur	PPQS	13	Kelas 2	8.0
Kamis	Muhammad Khatun	Anteroposterior Shoulder (2)	PPQS	17	Kelas 3	3.0
Jumlah	Lumutari	XXXX	PPQS	30	Kelas 2	11.0

Surabaya, 11 July 2018  
Panggung Jember

Lupa

Wednesday 11 July Page 2 of 2

### Jadwal Penggunaan Ruang Operasi RSUD Dr. Soetomo Divisi Ortopedi (Ruang 610)

hari	Nama Pasien	Jenis Kasus	Operator	Length of Stay (hari)	Kelas Prioritas	Lama Operasi (jam)
Senin	Tri Murni	AT Pilon Metatarsal (1)	PPQS	26	Kelas 2	8.0
Senin	Diana E	ORIF Scapula	PPQS	19	Kelas Utama	2.0
Senin	Sherafudin	AT Pilon Fraktur	PPQS	14	Kelas 4	1.0
Senin	Yusuf	Distraction Hip (R)	SEE	1	Kelas 1	2.0
Senin	M.A. Saad	Amputasi Zirkumfere	MED	18	Kelas 2	8.0
Senin	Ry. Nuzul	ORIF Proximal	PPQS	11	Kelas 2	3.0
Senin	Siti Dwi D.	AT Pilon + Scapula	PPQS	18	Kelas 2	2.0
Rabu	Arifin M. E.	Amputasi Above Knee	MED	17	Kelas Utama	3.0
Rabu	Theris	Operasi	PPQS	26	Kelas 1	1.0

Surabaya, 11 July 2018  
Panggung Jember

Lupa

Wednesday 11 July Page 1 of 2



Nam	Nama Pasien	Jenis Kasus	Operator	Length of Stay (hari)	Kelas Prioritas	Lama Operasi (Jam)
Radu	Dima	Stroke Focal + LO Focal	HSD	18	Kelas 3	9.0
Kami	Chaitanus S.	Arthroscopy Shoulder	SDY	28	Kelas Utama	3.0
Kami	Sin Wenging	Arthroscopy +	MDY	28	Kelas Utama	2.0
Kami	Nyia M.A.	ORIF Tibia Fracture	MCH	8	Kelas Utama	2.0
Kami	Aulika Mintha	Hemodialisa + Autogram	SBS	15	Kelas Utama	3.0
Jumi	Hu. Gus. Kurniati	AT Plantar Fascia	PPDS	22	Kelas 2	3.0
Jumi	Ny Hana Tahmawati	Stabilisasi Pankreas +	RPOS	21	Kelas 3	2.0

Surabaya: 14 July 2018  
Perangung JI web  
MIA

Jadwal Penggunaan Ruang Operasi RSUD Dr. Soetomo Divisi Ortopedi (Ruang 612)						
Nam	Nama Pasien	Jenis Kasus	Operator	Length of Stay (hari)	Kelas Prioritas	Lama Operasi (Jam)
Sani	Tu Ahmad Yudi	Distraction + Re ORIF	SDY	27	Kelas Utama	2.0
Sani	Tu Satrio	AT Ext. Fracture	MCH	24	Kelas Utama	2.0
Sani	Dinda E.	ORIF Scapula	PPDS	14	Kelas Utama	2.0
Sani	Tu Supen	Distraction Hip (R)	RDC	5	Kelas 3	2.0
Sani	Tu Mutiara	Distraction Spinal Focal	MDY	28	Kelas 3	2.0
Sani	Tama W.	Open Bunion Multiple	BSE	20	Kelas Utama	1.0
Sani	Tu Azzahra	ORIF Femur	PPDS	16	Kelas 2	2.0
Sani	Ragha Vanyana	Arthroscopy	SDY	17	Kelas 2	3.0

Surabaya: 14 July 2018  
Perangung JI web  
MIA

Thursday 14 July Page 1 of 2

Hari	Nama Pasien	Jenis Kasus	Operator	Length of Stay (Hari)	Kelas Prioritas	Lama Operasi (Jam)
Senin	An.Octelia B	Blount	JLP	15	Kelas 2	2.0
Rabu	An.Ismail E	Anaplast Alvea Nose	MED	27	Kelas Utama	3.0
Rabu	Th.Darmawati	PSR	IPDS	24	Kelas 2	1.0
Rabu	Ardani H.	Scal Plasticing	ARR	29	Kelas 2	3.0
Rabu	Ny.Wahyuni	DRF Aorticvalve	PPDS	19	Kelas 2	1.8
Kamis	Fahedy B	Steth (Hill + DeGroot)	MED	10	Kelas 2	2.0
Kamis	An.Arman Ari	Bonecortical Hip (D)	BDC	6	Kelas 3	1.0
Jumel	Lukman H.	PFMT	HSD	29	Kelas 2	4.0

Surabaya, 14 July 2016

PenanggungJawab

1000

Thursday 14 JulyPage 2 of 2

Jadwal Penggunaan Ruang Operasi  
RSUD Dr. Soetomo Divisi Ortopedi (Ruang 614)

Hari	Nama Pasien	Jenis Kasus	Operator	Length of Stay (Hari)	Kelas Prioritas	Lama Operasi (Jam)
Senin	Th.Manan	At Plst Medtarsal dig I	PPDS	28	Kelas 2	6.0
Senin	I Made Dwi K.	PFMT C5 Th.1	HSD	20	Kelas Utama	6.0
Selasa	Ny.Ngalining	Plating Aorticbukum	PPDS	27	Kelas 3	6.0
Selasa	An.Nabila	Tenotomy Adductor D/S	SBS	17	Kelas 3	3.0
Selasa	Nikmah	Wide Excisi + Lokal Flap	BEE	15	Kelas Utama	2.0
Rabu	Th.Suglio	Anterior Cervical	Prof.BP	27	Kelas 2	6.0
Rabu	Sella Dwi P.	At Plats + Screwing	PPDS	16	Kelas 2	2.0
Rabu	An.Mch.Widari	Osteotomy/Plating Humer	PPDS	15	Kelas Utama	3.0
Kamis	M.Ali Firmansjah	Revisi TENS Cruris (D)	SBS	24	Kelas 3	3.0

Surabaya, 14 July 2016

PenanggungJawab

1220

Thursday 14 JulyPage 1 of 2

Hari	Nama Pasien	Jenis Kasus	Operator	Length of Stay (Hari)	Kelas Prioritas	Lama Operasi (Jam)
Senin	Fitriana	Fracture Tibia	VPDS	23	Kelas 3	2.0
Senin	Edy Harnono	Tibia Fracture Traction	EDY	28	Kelas Utama	2.0
Senin	Sumarto	AR Tibia Fracture	VPDS	28	Kelas 2	1.8
<p>Saratnya: 14.8.2016</p> <p>Penanggung Jawab:</p> <p>Isa</p>						
<p>Thursday 14 July</p> <p>Page 2 of 2</p>						

Hasil penjadwalan untuk 49 pasien

<b>Jadwal Penggunaan Ruang Operasi</b> <b>RSUD Dr. Soetomo Divisi Ortopedi (Ruang 610)</b>						
Hari	Nama Pasien	Jenis Kasus	Operator	Length of Stay (Hari)	Kelas Prioritas	Lama Operasi (Jam)
Senin	Tn Mubarak	Debridement Surai Flap	MOY	28	Kelas 3	2.0
Senin	Tn Ahmad Yajid	Debridement + Re ORIF	EDY	27	Kelas Utama	2.0
Senin	I Made Dwi K.	FFMT C5 Th.1	HSO	20	Kelas Utama	6.0
Selasa	Andika Reza	Removal Ilizarov	JUP	28	Kelas Utama	3.0
Selasa	An Ahmad Anif	Disarticulasi Hip (D)	BEE	9	Kelas 3	2.0
Selasa	Ny Karfiri	Stabilisasi Posterior +	MCH	19	Kelas Utama	3.0
Selasa	Nikmah	Wide Excisi + Lokal Flap	BEE	15	Kelas Utama	2.0
Rabu	Ny Ngatining	Plating Acetabulum	PPDS	27	Kelas 3	6.0
Rabu	Dodik E.	ORIF Screwing	PPDS	19	Kelas Utama	2.0
<p>Saratnya: 14 July 2016</p> <p>Penanggung Jawab:</p> <p>Isa</p>						
<p>Thursday 14 July</p> <p>Page 1 of 2</p>						

Hari	Nama Pasien	Kerusakan	Operator	Length of Stay (Hari)	Kelas Prioritas	Lama Operasi (Jam)
Rabu	Soodbyo	AT TBW Palsia	PPDB	19	Kelas 2	1.0
Rabu	Muhah	Open Biopai Knee	BEE	14	Kelas Utama	2.0
Kamis	Dewi S.	Open Biopai	PPDB	26	Kelas 1	3.0
Kamis	Shi Wilijong	Arthroscopy +	MOY	26	Kelas Utama	2.0
Kamis	Ayuk Kristina	Excidi Tumor +	PAY	14	Kelas 3	6.0
Jumat	Lukman H.	FFMT	HBO	24	Kelas 2	6.0
Jumat	An.Mch.Wiklan	Ostectomy,Plating Herni	PPDB	15	Kelas Utama	3.0
Jumat	Atfudin	AT Interlocking Nail Tibia	PPDB	14	Kelas 2	1.0

Surabaya, 14 July 2016

**Penanggungjawab**

**izza**

Thursday 14 July

Page 2 of 2

<div> <div>Jadwal Penggunaan Ruang Operasi</div> <div>RSUD Dr. Soetomo Divisi Ortopedi (Ruang 612)</div> </div>						
Patient	Name Patient	Type Case	Operator	Length of Stay (Hari)	Room Prioritas	Room Operasi (Jam)
1	Chalimatus S.	Arthroscopy Shoulder	EDY	26	Kelas Utama	3.0
2	M.Ali Firmansjah	Revisi TENS Cruris (D)	SBS	24	Kelas 3	3.0
3	Ny.Hana Fatmawati	Stabilisasi Posterior +	PPDS	21	Kelas 3	2.0
4	An.Nabila	Tenotomy Adductor D/S	SBS	17	Kelas 3	3.0
5	Tn.Misran	All Plat Metatarsal dig.1	PPDS	28	Kelas 2	6.0
6	Dimas	Surral Flap + LD Flap	HSO	15	Kelas 3	6.0
7	An.Ismaina S.	Amputasi Above Knee	MED	27	Kelas Utama	3.0
8	Farady B.	Skin Graft + Debridement	MED	26	Kelas 3	2.0

Surabaya, 14 July 2018

Penanggung Jawab

Izza



Hari	Nama Pasien	Jenis Kasus	Operator	Length of Stay (Hari)	Kelas Prioritas	Lama Operasi (Jam)
Rabu	Ny. Juliah	FFMT	Teri	14	Kelas 3	6.0
Kamis	Tn. Turtaman	Af Ext.Fixasi	PPDS	25	Kelas 2	2.0
Kamis	Deddy Gunawan	Tibial Posterior Tendon	EDY	24	Kelas Utama	2.0
Kamis	Tn.Sukardi	TBW Patella	PPDS	18	Kelas 2	2.0
Kamis	Moedji Rahayu	Arthroscopy Shoulder (D)	MOY	17	Kelas 3	2.0
Jumat	Tn.Darmawan	PSR	PPDS	24	Kelas 2	1.0
Jumat	Emawati	Plating Tibia Plate	PPDS	23	Kelas 3	2.0

Surabaya, 14 July 2016  
Penanggung Jawab

Izza

---

Thursday 14 July Page 2 of 2

### Jadwal Penggunaan Ruang Operasi RSUD Dr. Soetomo Divisi Ortopedi (Ruang 614)

Hari	Nama Pasien	Jenis Kasus	Operator	Length of Stay (Hari)	Kelas Prioritas	Lama Operasi (Jam)
Senin	Tn.Slamet	Af Ext.Fixasi	MCH	24	Kelas Utama	2.0
Senin	Achmad N.	Serial Plastering	ARN	20	Kelas 2	2.0
Senin	Tarso W.	Open Biopsi Multiple	BEE	20	Kelas Utama	1.0
Senin	Yanto	Af Ext.Fixasi, Sham	PPDS	19	Kelas 2	2.0
Senin	Bagus Cahyono	Arthroscopy,	DNJ	17	Kelas 2	3.0
Selasa	Tn.M. Said	Amputasi Selinggi Bahu	MED	28	Kelas 2	6.0
Selasa	Aurora Mersha	Hemispica + Autogram	SBS	15	Kelas Utama	3.0
Selasa	Aulia Ananda	Hemispica + Autogram	SBS	14	Kelas Utama	2.0
Rabu	Tn.Suglo	Anterior Cervical	Prof.BP	27	Kelas 2	6.0

Surabaya, 14 July 2016  
Penanggung Jawab

Izza

---

Thursday 14 July Page 1 of 2

Hari	Nama Pasien	Jenis Kasus	Operator	Length of Stay (Hari)	Kelas Prioritas	Lama Operasi (Jam)
Rabu	Novi M.A.	ORIF Tibia Plateau	MCH	6	Kelas Utama	2.0
Rabu	An.Ghoffer R.	Ilizarov	JUP	15	Kelas 2	2.0
Kamis	An.Ahmad Sulaiman	Above Knee Amputasi	MED	9	Kelas 2	3.0
Kamis	Yusuf	Excisi Myositis	BEE	19	Kelas 2	3.0
Kamis	Ny.Wahyuni	ORIF Antebrachii	PPDS	17	Kelas 2	3.0
Kamis	Sella Dwi P.	At Plate + Screwing	PPDS	16	Kelas 2	2.0
Jumat	Ny.Siti Kodjajh	At Plate Fibula	PPDS	22	Kelas 3	3.0
Jumat	Th.Sapari	Disarticulasi Hrp (S)	BEE	5	Kelas 3	2.0
Surbaya, 14 July 2016 Penanggung Jawab  Izza						
Thursday 14 July				Page 2 of 2		

## **BAB VI**

### **PENUTUP**

Bab ini berisi tentang beberapa kesimpulan yang dihasilkan berdasarkan penelitian yang telah dilakukan. Disamping itu, pada bab ini juga dimasukkan beberapa saran yang dapat digunakan jika penelitian ini ingin dikembangkan.

#### **6.1 Kesimpulan**

Dari hasil pembahasan pada bab sebelumnya, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Dalam membuat sistem penjadwalan penggunaan ruang operasi RSUD Dr. Soetomo divisi bedah ortopedi dirumuskan 5 fungsi kendala yang terdiri atas urgensi, operator, *length of stay*, kelas prioritas, dan lama waktu operasi.
2. Optimalisasi algoritma genetika untuk menjadwalkan penggunaan ruang operasi dirumuskan dengan fungsi *fitness* yang merupakan gabungan antara fungsi objektif dan fungsi kendala yang telah didefinisikan pada bab sebelumnya.
3. Telah dirancang dan dibuat perangkat lunak penjadwalan penggunaan ruang operasi RSUD Dr. Soetomo divisi ortopedi yang menghasilkan jadwal yang optimal.

## **6.2 Saran**

Dari hasil capaian penelitian yang telah dilakukan, ada beberapa hal yang penulis sarankan diantaranya:

1. Algoritma genetika dapat dikembangkan lagi dengan mempertimbangkan perkiraan lama operasi yang dapat berubah sewaktu-waktu.
2. Algoritma genetika ini dapat digunakan untuk membuat penjadwalan dalam permasalahan yang lain.

## DAFTAR PUSTAKA

- Chang, P., Chen, S., & Chan, C.-L. (2008). Genetic algorithm integrated with artificial chromosomes for multi-objective flowshop scheduling problems. *Applied Mathematics and Computation*, 550-561.
- Firmansyah, A., Irawan, M. I., & Utomo, D. B. (2016). ALGORITMA GENETIKA DENGAN MODIFIKASI KROMOSOM UNTUK PENYELESAIAN MASALAH PENJADWALAN FLOWSHOP. *JURNAL SAINS DAN SENI ITS*, 1-6.
- Hozairi, A., K. B., Masroeri, & Irawan, M. I. (2014). IMPLEMENTATION OF NONDOMINATED SORTING GENETIC ALGORITHM -- II (NSGA-II) FOR MULTIOBJECTIVE OPTIMIZATION PROBLEMS ON DISTRIBUTION OF INDONESIAN NAVY WARSHIP. *Journal of Theoretical & Applied Information Technology*, 274-281.
- Hozairi, Artana, K. B., Masroeri, & Irawan, M. I. (2012). IMPLEMENTATION OF INTELLIGENT CONTROL FOR OPTIMIZATION OF FLEET PLACEMENT TNI-AL SHIPS USING GENETIC ALGORITHM. *Academic Research International*, 17-30.
- Kusumadewi, S. (2005). *Penyelesaian Masalah Optimasi dengan Teknik-Teknik Heuristik*. Yogyakarta: Graha Ilmu.

- Li, Y., & Chen, Y. (2010). A Genetic Algorithm for Job-Shop Scheduling. *Journal of Software*, Vol. 5, No. 3, 269-274.
- Muhtaromi, M., & Irawan, M. I. (2015). Perancangan Prototipe Perangkat Lunak untuk Penempatan Pegawai dengan Model Pilihan dari Perspektif Dua Arah Berbasis Algoritma Genetika. *JURNAL SAINS DAN SENI ITS Vol. 4, No.2*, A73-A78.
- Pudji, S., Artana, K. B., Masroeri, A., Dinariyana, A., & Irawan, M. I. (2014). IMPLEMENTASI BINARY GENETIC ALGORITHM (BGA) UNTUK OPTIMASI PENUGASAN KAPAL PATROLI TNI - AL DALAM RANGKA KEAMANAN WILAYAH LAUT INDONESIA. *Seminar Nasional IENACO-2014*, 398-407.
- RSU Dr Soetomo. (n.d.). Retrieved from Find The Best: <http://rumah-sakit.findthebest.co.id/l/232/RSU-Dr-Soetomo>
- Santoso, P., A., K. B., Masroeri, Irawan, M. I., & Dinariyana, A. (2014). THE IMPLEMENTATION OF BINARY GENETIC ALGORITHM (BGA) FOR OPTIMIZING THE TASK OF INDONESIAN NAVY SHIP PATROLS RELATED TO THE SECURITY OF INDONESIA SEAS. *Journal of Theoretical & Applied Information Technology*, 247-253.
- Shahab, M. L., Utomo, D. B., & Irawan, M. I. (2016). DECOMPOSING AND SOLVING CAPACITATED VEHICLE ROUTING PROBLEM (CVRP) USING TWO-STEP GENETIC ALGORITHM (TSGA). *Journal of*

*Theoretical and Applied Information Technology*, 461-468.

- Sivanandam, S., & Deepa, S. (2008). *Introduction to Genetic Algorithms*. New York: Springer.
- Tormos, P., Lova, A., Barber, F., Ingolotti, L., Abril, M., & Salido, M. (2008). A Genetic Algorithm for Railway Scheduling Problems. *Springer*, 255-276.
- Wati, D. A. (2010). *Sistem Kendali Cerdas*. Yogyakarta: Graha Ilmu. Retrieved from <http://rumah-sakit.findthebest.co.id/1/232/RSU-Dr-Soetomo>
- Xiang, W., Yin, J., & Lim, G. (2015). An ant colony optimization approach for solving an operating room surgery scheduling problem. *Computers & Industrial Engineering*, 335-345.
- Zhao, Z., & Li, X. (2014). Scheduling elective surgeries with sequence-dependent setup times to multiple operating rooms using constraint programming. *Operations Research for Health Care*, 160-167.





## BIODATA PENULIS



Penulis bernama Lailatul Izza atau yang biasa dipanggil dengan Izza. Penulis dilahirkan di Sidoarjo, 11 September 1993. Penulis merupakan putri pertama dari pasangan Bapak Suriyadi dan Ibu Maslakhah. Penulis menempuh pendidikan di RA Bahrul Ulum, SDN Becirongengor, SMP Negeri 1

Sidoarjo, dan SMA Negeri 4 Sidoarjo.

Kemudian penulis melanjutkan studi di Jurusan Matematika ITS pada tahun 2012 untuk menempuh pendidikan S1 Matematika dengan NRP 1212 100 023. Di Jurusan Matematika ITS, penulis mengambil bidang minat Ilmu Komputer. Selama kuliah penulis juga mengikuti kegiatan organisasi yaitu aktif di Himpunan Mahasiswa Jurusan Matematika ITS (HIMATIKA ITS). Pada tahun periode 2013 – 2014 penulis menjadi staff Departemen Sosial Masyarakat HIMATIKA ITS dan pada tahun periode 2014 – 2015 penulis menjabat sebagai Kepala Departemen Pengabdian Masyarakat HIMATIKA ITS. Selain aktif dalam organisasi, penulis aktif mengikuti kepanitiaan berbagai acara, seperti GERIGI ITS, OMITS, ISG SOSDEV dan Gathering Alumni Mahasiswa Matematika ITS.

Untuk informasi lebih lanjut dan jika ingin memberikan saran Tugas Akhir ini bisa ditujukan ke penulis melalui email *lailatulizza106@gmail.com*.